UNIVERSITÀ degli STUDI di CATANIA | DIPARTIMENTO DI INGEGNERIA ELETTRICA, ELETTRONICA E INFORMATICA

DOTTORATO DI RICERCA IN INGEGNERIA DEI SISTEMI,
ENERGETICA, INFORMATICA E DELLE TELECOMUNICAZIONI
XXXI CICLO

Ph.D. Thesis

DEEPLY INCORPORATING HUMAN CAPABILITIES INTO MACHINE LEARNING MODELS FOR FINE-GRAINED VISUAL CATEGORIZATION

ING. FRANCESCA MURABITO

Coordinatore	Tutor
Chiar.mo Prof.	Chiar.mo Prof.
P. ARENA	C. SPAMPINATO

alla mia famiglia, per l'amore incondizionato
a Simone, per avermi ispirata
a Concetto, per avermi guidata

ABSTRACT

Artificial intelligence and machine learning have long attempted to emulate human visual system. With the recent advances in deep neural networks, which take inspiration from the architecture of the primate visual hierarchy, human-level visual abilities are now coming within reach of artificial systems. However, the existing computational models are designed with engineering goals, loosely emulating computations and connections of biological neurons, especially in terms of intermediate visual representations.

In this thesis we aim at investigating how human skills can be integrated into computational models in order to perform fine-grained image categorization, a task which requires the application of specific perceptive and cognitive abilities to be solved. In particular, our goal is to develop systems which, either implicitly or explicitly, combine human reasoning processes with deep classification models. Our claims is that by the emulation of the process carried out by humans while performing a recognition task it is possible to yield improved classification performance.

To this end, we first attempt to replicate human visual attention by modeling a saliency detection system able to emulate the integration of the top-down (task-controlled, classification-driven) and bottom-up (sensory information) processes; thus, the generated saliency maps are able to represent implicitly the way humans perceive and focus their attention while performing recognition, and, therefore, a useful supervision for the automatic classification system. We then investigate if and to what extent the learned saliency maps can support visual classification in nontrivial cases. To achieve this, we propose *SalClassNet*, a CNN framework consisting of two networks jointly trained: a) the first one computing top-down saliency maps from input images, and b) the second one exploiting the computed saliency maps for visual classification.

Gaze shifts change in relation to a task is not the only process when performing classification in specific domains, but humans also leverage a-priori specialized knowledge to perform recognition. For example, distinguishing between different dog breeds or fruit varieties requires skills that not all human possess but only domain experts. Of course, one may argue that the typical *learning-by-example* approach can be applied by asking domain experts to collect enough annotations from which machine learning methods can derive the features necessary for the classification. Nevertheless, this is a really costly process and often infeasible. Thus, the second part of this thesis aim at explicitly modeling and exploiting domain-specific knowledge to perform recognition.

To this end, we introduce and demonstrate that computational ontologies can explicitly encode human knowledge and that it can be

used to support multiple tasks from data annotation to classification. In particular, we propose an ontology-based annotation tool, able to reduce significantly the efforts to collect highly-specialized labels and demonstrate its effectiveness building the *VegImage* dataset, a collection of about 4,000 images belonging to 24 fruit varieties, annotated with over 65,000 bounding boxes and enriched with a large knowledge base consisting of more than 1,000,000 OWL triples.

We then exploit this ontology-structured knowledge by combining a semantic-classifier, which performs inference based on the information encoded in the domain ontology, with a visual convolutional neural network, showing that the integration of semantics into automatic classification models can represents the key to solve a complex task such as the fine-grained recognition of fruit varieties, a task which requires the contribution of domain expert to be completely solved.

Performance evaluation of the proposed approaches provides a basis to assess the validity of our claim along with the scientific soundness of developed models, able to effectively integrate human skills in the visual categorization process.

CONTENTS

CHAPTER

ONE

MOTIVATION AND OBJECTIVES

The fascinating idea of an artificial intelligence system able to perceive, think and act in the same way a human being does, has been a recurrent topic in the contemporary culture, ranging from romances to movies, where the plot is mostly focused on the extend of the feelings a machine can experience, to science blogs, which are usually more interested on how the newly created intelligence may affect the everyday life. While the aspect of machine feelings has been relegated to science-fiction authors' imagination, researchers have been working on developing algorithms which could help people in performing everyday task. An example is the recently introduced *Google Duplex*, an AI System able to have a conversation over the phone in order to accomplish real-world tasks such as scheduling a hair salon appointment or calling a restaurant, totally replacing the human operator. Google researchers have pushed the limits of machine learning capabilities with the *Magenta* project, a collection of algorithms designed for art and music production, thus attempting to emulate the human creative process. Human creativity potentialities are particularly expressed in lifelong learning experiences, where humans are embedded in a cycle *Perception-Elaboration-Memorization* of new information in order to develop structured knowledge and skills which are then reused in similar tasks. Even though machines are still far from being able to perform tasks they have not been trained for, their performance on well defined scenarios have reached impressive performance in the last years, specially since the rediscovery of *Convolutional Neural Networks* applied to solve computer vision problems such as object recognition or object localization [1, 2, 3, 4]. Whilst AI algorithms become more and more sophisticated, with machines, for example, able to beat human champions at chess, Go and Jeopardy, recently Uber has announced that self-driving cars will be back on public roads, but under human control, thus leaving open the debate on what should be the human contribution essential for a safe interaction with AI systems. Despite not exploring these limits, which would even pose moral and ethical issues, in this thesis we aim at investigating how computation models can encode human skills and whether these skills may help in improving accuracy of computer vision methods. In particular, we develop machine learning models which incorporate, either implicitly or explicitly, the capabilities humans have and the knowledge they exploit in performing visual categorization. To this end, we study how human knowledge can support fine-grained classification, which is a typical problem requiring specific perceptive skills and reasoning processes, starting from a very implicit way of providing human contribution (i.e., through saliency maps which emulate the human ability to focus on discriminative regions), then demonstrating how computational ontologies are able to explicitly encode structured human knowledge and boost the process of dataset annotation and finally proving that through the exploitation of this structured knowledge it is possible to improve fine-grained classification accuracy.

Indeed, computer vision and machine learning methods have long attempted to emulate humans while performing visual tasks. Despite the high intentions, the majority of the existing automated methods rely on a common schema, i.e., learning low- and mid-level visual features for a given task, often without taking into account the peculiarities of the task itself. One of the most relevant example of task-driven human process is visual attention, i.e., gating visual information to be processed by the brain according to the intrinsic visual characteristics of scenes (bottom-up process) and to the task to be performed (top-down process). Saliency detection building only on the bottom-up process mainly employs low-level visual cues, modeling unconscious vision mechanisms, and shows huge limitations in task-oriented computer vision methods. For example, traditional saliency methods (5) miss objects of interest in highly cluttered backgrounds since they detect visual stimuli, which often are unrelated to the task to be accomplished, as shown in Figure 1.1. Analogously, image classifiers fail in cases of cluttered images as they tend to extract low and mid-level visual descriptors and match them with learned data distributions without focusing on the most salient image parts.

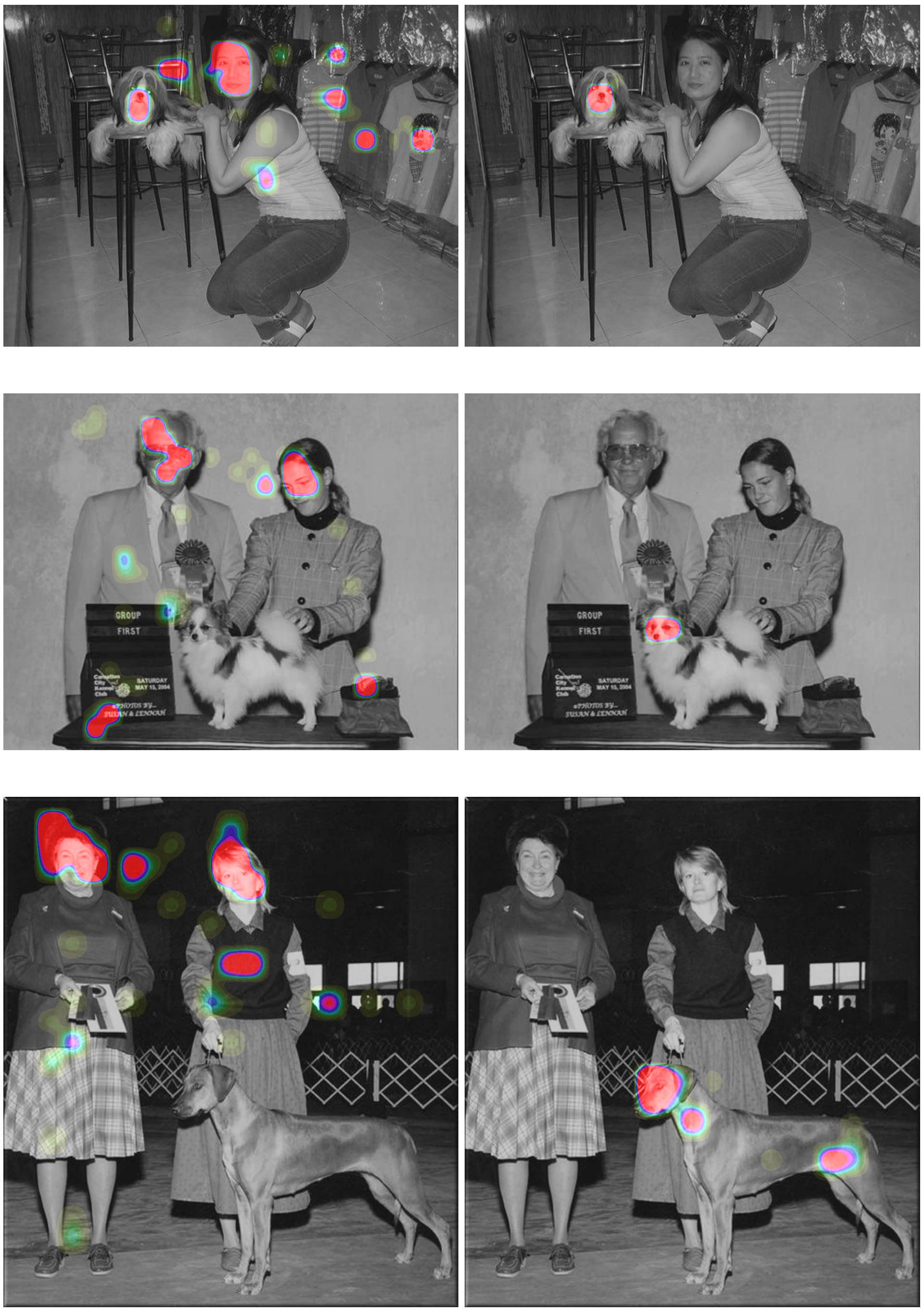


Figure 1.1: **First column** — *Eye fixations in free-viewing experiments in images with multiple objects. Some of the salient regions cannot be used for dog species classification.* **Second Column** — *Eye fixation shifts when asking to guess dog breeds.*

Under this scenario, we investigate whether it is possible to learn saliency maps encoding the human ability to identify task-dependent discriminative regions, i.e., those regions that could help in performing fine-grained recognition. Thus, our main objective is to develop a method for saliency detection guided by a classification task and demonstrate that exploiting task-based saliency maps improves classification performance. Since it is not possible to imagine an automatic system using human data, especially in inference, we we propose and train, in an end-to-end fashion, a convolutional neural network — *SalClassNet* — consisting of two parts: the first one generating top-down (classification-guided) saliency maps from input images, while the second one taking images and the learned maps as input to perform visual categorization (see Chapter 3 for more details). In particular, we demonstrate how the propagation of a mixed saliency/classification loss throughout the upstream SalClassNet saliency detector is the key to learn task-guided saliency maps able to better detect the most discriminative features in the categorization process.

If the integration of saliency maps, computed emulating the way humans focus their attention while performing classification, can improve recognition, we want to investigate whether it is possible to emulate the complex perceptive, cognitive and reasoning processes humans carry out when a classification task has to be accomplished. Indeed, in spite of their impressive performance on the 1000-class ImageNet dataset, classification approaches are still predominantly based on visual features, whose distributions are learned through machine learning algorithms. While this has proved to be an effective strategy even in complex scenarios such as fine-grained image classification [6, 7, 8], there are cases where relying on visual appearance only might fail, especially in specific application domains (for which it is very complex — if not impossible — to collect large scale samples). For example, Figure 1.2 (left) shows four different varieties of cherry ("Bing", "Black Tartarian", "Burlat" and "Lapin") that cannot be easily discriminated through typical visual description approaches. However, objects in the real world do not appear as isolated items, but come in a rich context (the right-hand image in Figure 1.2 shows the same cherry varieties in their natural context), which is largely exploited by human experts to infer visual classes.

We believe that research in computer vision should also look into this direction, especially in all those cases where visual cues alone fail to express the difference among classes especially when the limited number of images hinders the capabilities of deep learning methods to learn class distributions. The need for exhaustive knowledge is highlighted by the recent sprout of methods employing high-level knowledge (mainly unstructured) for computer vision tasks: knowledge transfer methods [9, 10, 11] and semantic relation graphs [12, 13] have been adopted to deal with the limitations of traditional multi-class or binary models, which suffer from being overly restrictive or overly relaxed, respectively. Our hypothesis is that, for a real breakthrough in computer vision, computers need to emulate the human visual process by combining perceptive elements (visual descriptors) and cognitive-semantic factors (high-level knowledge). However, employing high-level semantic information into computer vision methods is not trivial and poses several challenges: *How can we extract and represent visual world semantics, especially for very specific applica-*

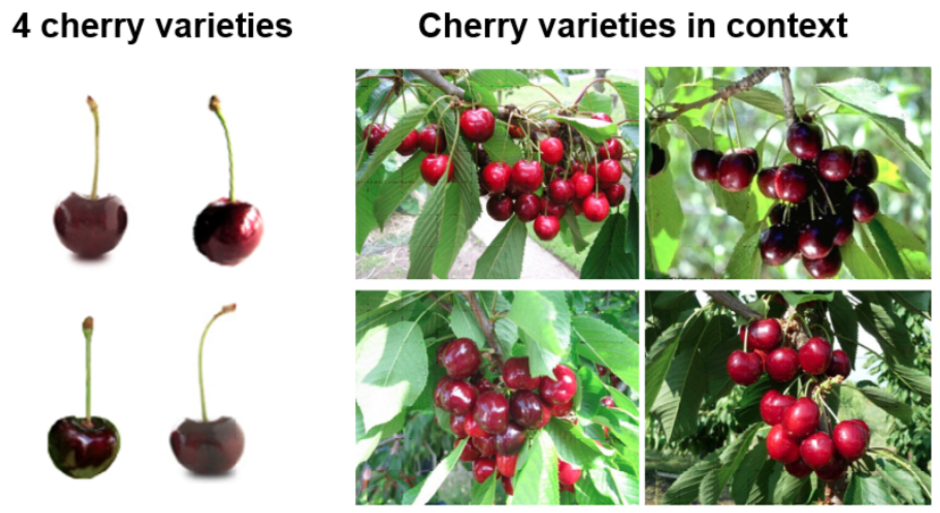


Figure 1.2: *Left: four cherry varieties, namely (left to right, top to bottom) "Bing", "Black Tartarian", "Burlat" and "Lapin". Right: the same varieties in their natural context. Information about leaf shape, distance between fruit and leaves, peduncle length may support the disambiguation between the four object classes.*

tion domains? How can we effectively integrate high-level semantic information into machine learning methods? How do we build computer vision systems tackling extremely complex tasks with a limited number of samples?

We aim at addressing the above questions by proposing a new "human in the loop" approach, where humans are not seen as simple workers, as in the classic sense of the paradigm [14, 15], but rather as the main core of the system, providing the necessary intelligence that is then encoded into machines to make them perform specific visual tasks. More specifically, our objectives are:

- adopting computational ontologies as a principled way to model — in a machine readable manner — visual world semantics as perceived by humans and showing how they can guide the visual annotation process according to a specialized domain knowledge model.
- leveraging high-level structured knowledge for devising a classification approach able to combine low level visual cues and semantic information to solve complex vision tasks.

As a test applications, for all research branches, we tackle the problem of fine-grained image classification in a complex tasks which require extensive human-knowledge to be solved: fruit variety image classification.

The comparison between our approach and state-of-the-art deep learning methods, over the considered scenario, shows a performance improvement when employing structured semantics. Thus, Chapter 4 introduces our domain-agnostic ontology and its generalization capabilities in modeling visual knowledge for specific application domains; then, in order to to prove computational ontologies' potentialities, we present our ontology-based annotation tool and show how it can boost the annotation process in two different scenarios: the creation of *VegImage*, a new fine-grained image dataset enriched with structured knowledge, and the curation of photographic data and text documents of historical buildings for indexing, retrieval and classification through a specific extension of the annotation tool, named *CulTO*. Afterwards, Chapter 5 describes how a classifier, able to take advantage of ontology-encoded domain knowledge, can support a standard visual convolutional classifier, showing also the results achieved by our knowledge-powered classifier on the test dataset.

Finally, Chapter 6 draws the conclusions of this thesis, pointing out the importance of the integration, whether implicit or explicit, of human skills to solve computer vision tasks.

CHAPTER TWO

LITERATURE REVIEW

This Chapter provides the reader with the state of the art on the topics covered in this thesis. Since the main objective is to transfer both perceptive, through saliency maps, and cognitive, by the integration of structured knowledge, human skills to computational models for fine-grained visual classification, we first present recent saliency detection methods. Afterwards, we introduce current computer vision methods exploiting semantic visual information and the advantages of ontologies in representing semantics through structured knowledge; then, the use of ontologies in cultural heritage is presented too in order to show their potentialities in disparate domain applications. Finally, we describe how saliency maps and semantics have been exploited to perform fine-grained visual classification.

2.1 Saliency Detection

Visual attention in humans can be seen as the integration between a) an early bottom-up unconscious process where the attention is principally guided by some coarse visual stimuli, which can be local (e.g., center-surround mechanisms) or global (dependent from the context); and b) a late top-down process, which biases the observation towards those regions that consciously attract users' attention according to a specific visual task. While the former has been extensively researched in the computer vision field with a significant number of proposed saliency detection methods ([16, 17, 18, 19, 20, 21, 22, 23, 24]), top-down processes have received much less attention ([25, 26, 27, 28]), mainly because of the greater difficulty to emulate high-level cognitive processes than low-level cues based on orientation, intensity and color ([5]).

Recently, the rediscovery of convolutional neural networks and their high performance on visual tasks have led to the development of deep saliency detection networks that either adopt multi-scale patches for global/coarser and local/finer features extraction for further saliency assessment (20, 21, 23, 22, 29, 24, 16, 18, 21, 30, 31, 23, 32, 33) or learn, in an end-to-end fashion, saliency maps as in [18, 19, 34]. In particular, the recent work by [19] presents a fully-convolutional CNN (partly trained from scratch and partly re-using low-level layers from existing models) for saliency prediction; another fully-convolutional architecture is the one presented in [18], which processes images at two different scales and is based on deep neural networks trained for object recognition; the latter was used as basis for our work as described later.

Recent saliency detection methods have been fed with high level information in order to include top-down attention processes. In [35], given the class label as prior, the parameters of a new feedback layer are learned to optimize the target neuron output by filtering out noisy signals; in [36] a new backpropagation scheme, "Excitation Backprop", based on a probabilistic version of the Winner-Take-All principle, is introduced to identify task-relevant neurons for weakly-supervised localization. Our saliency maps differ from the ones computed by those methods since the only top-down signal introduced in our training is a class-agnostic classification loss; hence, our maps are able to highlight those areas which are relevant for classifying generic images.

Given that our idea is to create an end-to-end model where the computed saliency maps are directly fed into the classification network, the most interesting saliency detection architectures for our purpose are the fully-convolutional ones, whose output can be seamlessly integrated into a larger framework with a cascading classification module. Tab. 2.1 summarizes the results of state-of-the-art fully-convolutional saliency networks on a set of commonly-employed datasets for saliency detection benchmarking, namely SALICON validation and test sets ([37]), iSUN validation and test sets ([38]) and MIT300 ([39]).

Method	**N. Layers**	**Framework**	**Training Dataset**	**SALICON Test**	**SALICON Val**	**iSUN Test**	**iSUN Val**	**MIT300**	
JuntingNet	5	Lasagne	SALICON	CC = 0.60 Shuffled-AUC = 0.67 AUC Borji = 0.83	CC = 0.58 Shuffled-AUC = 0.67 AUC Borji = 0.83	CC = 0.82 Shuffled-AUC = 0.67 AUC Borji = 0.85	CC = 0.59 Shuffled-AUC = 0.64 AUC Borji = 0.79	CC = 0.53 Shuffled-AUC = 0.64 AUC Borji = 0.78	
SalNet	10	Caffe	SALICON	CC = 0.62 Shuffled-AUC = 0.72 AUC Borji = 0.86	CC = 0.61 Shuffled-AUC = 0.73 AUC Borji = 0.86	CC = 0.62 Shuffled-AUC = 0.72 AUC Borji = 0.86	CC = 0.53 Shuffled-AUC = 0.63 AUC Borji = 0.80	CC = 0.58 Shuffled-AUC = 0.69 AUC Borji = 0.82	
SALICON	16	Caffe	OSIE	—	—	—	—	CC = 0.74 Shuffled-AUC = 0.74 AUC Borji = 0.85	
DeepGaze	5	Not Available	MIT1003	—	—	—	—	CC = 0.48 Shuffled-AUC = 0.66 AUC Borji = 0.83	
DeepGaze 2	19	Web Service	SALICON - MIT1003	—	—	—	CC = 0.51 Shuffled-AUC = 0.77 AUC Borji = 0.86		
ML-NET	19 + 2	Theano	SALICON	CC = 0.76 Shuffled-AUC = 0.78	—	—	—	—	CC = 0.69 Shuffled-AUC = 0.70 AUC Borji = 0.77
DeepFix	20	Not Available	SALICON	—	—	—	—	CC = 0.78 Shuffled-AUC = 0.71 AUC Borji = 0.80	
eDN	Ensemble	Sthor	MIT1003	—	—	—	—	CC = 0.45 Shuffled-AUC = 0.62 AUC Borji = 0.81	
PDP	16 + 3	Not Available	SALICON	CC = 0.77 Shuffled-AUC = 0.78 AUC Borji = 0.88	CC = 0.74 Shuffled-AUC = 0.78	—	—	CC = 0.70 Shuffled-AUC = 0.73 AUC Borji = 0.80	

Table 2.1: *A summary of state-of-art fully-convolutional methods and their results, according to the most common metrics, on several saliency datasets. Dataset references:* ***SALICON Test and Val****: [37];* ***iSUN Test and Val****: [38];* ***MIT300****: [39]. Method references:* ***JuntingNet and SalNet****: [19];* ***SALICON****: [18];* ***DeepGaze****: [17];* ***DeepGaze2****: [40];* ***ML-NET****: [41]:* ***DeepFix****: [42];* ***eDN****: [43];* ***PDP****: [44].*

.

2.2 Semantics in Computer Vision

Recently, there have been significant advances in modeling rich semantics using contextual relationships [45, 46, 47, 48, 49] (such as object-object [50, 46, 48] and object-attribute [51, 52, 53]) applied to scene classification [54], label propagation [55],object recognition [56] object detection [57] and zero-shot learning [58]. In [56], the authors showed that context information is more relevant for object recognition than the intrinsic object representation, especially in cases of poor image quality. However, visual scenes provide richer semantics than object-object or object-attribute relationships, which most of the existing methods do not take into account or do not exploit effectively as they try to solve the recognition problem by brute force learning.

Nevertheless, one of the limitations to a larger use of high-level knowledge in computer vision is the lack of structured resources for exhaustively modeling the semantics of our visual world. Indeed, until recently, the largest resource of structured visual knowledge is the ImageNet dataset [59], which, however, captures only limited semantic relations, ignoring, for instance, co-occurrence, dependency, mutual exclusion. Lately, an attempt to provide a dataset designed tor cognitive tasks is represented by Visual Genome [60], a collection of over 109K images where each image has an average of 35 objects, 26 attributes and 21 pairwise relationships between objects.

Moreover, including high-level knowledge in the learning loop is not trivial because of the complexity to make machine learning methods work with heterogeneous information at different abstraction levels. Recently, graph-structured representations have provided an effective way to encode semantic visual knowledge into machine learning approaches, and as such, have gained a lot of interest to describe semantically visual scenes [61, 62, 63, 12, 64]. For example, [64, 12] learned to generate graphs from images through Conditional Random Fields (CRF). Yet, given the high variability of semantic visual descriptions, it is extremely complex and time-consuming to identify the graph best fitting a given image, because of the large topology variability that image grounded graphs may have, beside showing low generalization capabilities.

To overcome these limitations, we propose to use computational ontologies for modeling visual world semantics. The two main advantages of computation ontologies over other semantic representations are:

- **Generalization capabilities.** By definition, computational ontologies, written through standard formalisms such as OWL, can be extended at will and integrated in other ontologies (and so can ontology instances) and other sources of semantic information (e.g., DBpedia, which, for instance, provides structured description of fruit habitat, useful for classification). This is done without any specific effort by data curators and without even the need to move data (since OWL supports web links), thus making it also a powerful tool for generating distributed and scalable datasets. Under this perspective, ontologies can be seen as strategic resources for creating common and standard visual distributed benchmarks that can keep increasing in size with low curation costs (this is not the case of ImageNet, for instance, since only ImageNet people can curate it).

- **Deep domain modeling.** Past approaches, such as scene graphs [64], tried to incorporate knowledge into automated methods, but they adopted simple semantics, ignoring complex relations which often occur in our visual world, such as mutual exclusion and inclusion. All these relations can be modeled through ontologies. Besides, computational ontologies allow for imposing constraints (e.g., "if the *Reinette* apple has red overcolour it cannot have russet") about real-world object appearance and their semantic relations.

Thus, computational ontologies are able to generalize semantic visual representations by describing, in an unified and standard framework, different types of properties, such as object attributes, object co-occurrence, hierarchy, pairwise object relationships, etc.

However, despite their great advantages over other semantic representations, computational ontologies have been largely underexploited in computer vision, except for some sporadic attempts [65], where human operators had to manually link ontology concepts to low-level image representations. This turned out to be more complex than expected and was one of the main reasons why these methods were soon abandoned.

Besides, another difficulty when dealing with computer vision problems, specially in the case of fine-grained classification, is the generation of large and correct visual annotations. This process is made more complex when visual data should be enriched with semantic and a priori-knowledge about objects, such as color, shape, related-objects and their visual properties, etc.. This, especially, holds in specialized application domains (e.g., fruit variety, bird, medical images, cultural heritage, etc.) where high precision is necessary to avoid affecting the learning process. In such cases, annotations should be provided by domain experts [14, 66], who are very expensive, and not always available, resources.

Computational ontologies are able to support computer vision scientists also in this regard, highlighting, once more, the support that such tools may provide to the research in the field. Indeed, high-level visual knowledge encoded in ontologies can be exploited to guide/constrain the annotation process and to infer visual attributes through ontology reasoning, hence reducing greatly the knowledge required to carry out the task. Thus far, only few methods exploiting this idea have been proposed [67, 68, 69, 70], and were mainly devised for information retrieval rather than computer vision.

Accordingly, we generate knowledge-enriched visual annotations on fruit variety images, thus providing a complex benchmark for fine-grained recognition [71]. There are several benchmarking datasets for fine-grained classification of birds, stonefly larvae, etc. [7**?** , 6] but they mainly contain per-instance segmentations and do to provide any semantic visual descriptions of objects and their context. The datasets most similar to ours are the ones for semantic scene labeling [72, 73], which, despite including context information, no exhaustive semantic relations are defined.

Ontology instead have found fertile ground in the cultural heritage because of the need to integrate, enrich, annotate and share the produced data [74, 75, 76, 77]. Cultural heritage ontologies are often employed to support the development of high-level software tools for digital content exploitation. In [78], a web-based virtual museum based on ontology is proposed where visitors can perform queries and create shared information by adding textual annotations. These new generation of approaches has enabled the conversion of traditional cultural heritage website into a well-designed and more content-rich one [79], integrating distributed and heterogeneous resources, thus overcoming the limitations of systems such as MultimediaN E-Culture project [80], which, instead, manually performs data enchriment through semantic web techniques for harvesting and aligning existing vocabularies and metadata schemas. MultimediaN E-Culture project also developed a new software, named “ClioPatra”, which allows users to submit queries based on familiar and simple keywords.

One of the biggest challenge that the information retrieval in the Cultural Heritage domain has to face is the natural heterogeneity of data. One of the main attempts to provide a unified access to digital collections is the “CatchUp” full-text retrieval system [81]. Ontologies have been often exploited in image retrieval systems to improve accuracy as they allow for bridging the “semantic gap”, i.e., the gap between the low-level content-based features and the data interpretation given by users. In the “eCHASE” project [82] several cultural heritage institution metadata schemas have been mapped into the CIDOM/CRM to expose them using the “Search and Retrieve Web Service” (SRW). Recently the “INCEPTION” project [83] has been focused on the innovation in 3D modelling of cultural heritage assets, enriched by semantic information, and their integration in a new H-BIM. The peculiarity of the system is that users are able to query the database using keywords and visualize a list of H-BIM models, description, historic information and the corresponding images, classified through the application of deep learning techniques.

2.3 Human-based Visual Classification Systems

In this section we show how visual classification algorithms have attempted to exploit, as a direct feedback or trying indirectly to emulate their potentialities, human skills in order to improve their accuracy. In particular, we present classification methods relying on saliency maps to identify the correct label that has be assigned to an image; from this point of view, saliency maps can be considered as a guideline suggesting which areas the algorithms should focus on . Afterwards, we introduce the fine-grained classification problem along with the issues related to its solution; in order to overcome those issues researchers have tried to combine visual features with high-level semantics which explicitly represent the way humans perceive and reason.

2.3.1 Saliency for Visual Classification

Understanding the processes which are behind task-controlled visual attention may be of crucial importance to make machines see and understand the visual world as humans do and to solve complex vision tasks, such as recognition of multiple objects in cluttered scenes ([84]).
The idea of using saliency for improving classification performance has gained significant attention from the computer vision community, coming up with saliency detection models that have been integrated into visual classification methods. In [85], saliency maps are employed to weigh features both in the learning and in the representation steps of a sparse coding process, whereas in [86] CNN-based part detections

are encoded via Fisher Vectors and the importance of each descriptor is assigned through a saliency map. [87] extended the recurrent attention model (RAM) presented in [88] (a model based on a combination between recurrent neural networks and reinforcement learning to identify *glimpse* locations) by training it to detect and classify objects after identifying a fixed number of glimpses.

A work similar in the spirit to ours is [89], where a low-capacity network initially scans the input image to locate salient regions using a gradient entropy with respect to feature vectors; then, a high-capacity network is applied to the most salient regions and, finally, the two networks are combined through their top layers in order to classify the input image. Our objective, however, is to perform end-to-end training, so that the classification error gradient can directly affect the saliency generation process.

2.3.2 Semantics in Fine-grained Recognition

The importance of visual world semantics (and of context, especially) in automated visual recognition has been long acknowledged [90, 50]. As mentioned in the Chapter 1, we apply our classification methods for fine-grained visual recognition tasks, which particularly demand for a priori-knowledge about objects, their visual appearance and their context given the high similarity among classes. Indeed, fine-grained image classification is one of those visual tasks on which classification approaches, even based on deep-learning, have shown major difficulties. Most of the first classification methods tried to identify and learn — using sometimes simple semantics as in [91] — distributions of visual descriptors [92] and object parts [93, 94], often using deformable part models [95] and requiring pose normalization techniques [93, 96, 95] to align object appearances. Recently, deep learning has been employed to improve fine-grained classification performance. In [97], deep-learned features, computed on an augmented (through component-wise power transform) image dataset, are used to perform recognition, while the authors of [98] embedded descriptors based on convolutional neural network (CNN) weights (pre-trained on ImageNet) in the deformable part descriptors (DPD) model; a bilinear CNN model is introduced in [99] to capture pairwise correlations between the feature channels and represent part-feature interactions. Furthermore, CNNs have been largely exploited to tackle pose-normalization [100, 101, 102], semantic part detection for fine-grained recognition [103] or unsupervised part discovery issues [104], and those methods have sensibly increased the performance on complex fine-grained datasets. Nevertheless, all these approaches are not able to exploit the semantics of the scene, showing evident difficulties when information other than visual ones has to be employed. An attempt to exploit rich professional knowledge for fine-grained visual recognition is proposed in the recent work [105]: their KERL framework organizes knowledge about categories and part-based attributes in the form of a knowledge graph taken as input by a Gated Graph Neural Network [106] to learn a representation which is then embedded into the image feature learning to extract attribute-aware features; while their GGNN implicitly associates specific attributes with feature maps, we explicitly link objects in a image to the semantic relations and attributes defined in the ontology for that object.

CHAPTER

THREE

TOP-DOWN SALIENCY DETECTION DRIVEN BY VISUAL CLASSIFICATION

In this Chapter we present *SalClassNet*, an end-to-end convolutional neural networks consting of two modules: a) a saliency detection network which tries to emulate the way humans focus their attention while performing a classification task and generates saliency maps suitable to be exploited by b) a classification network in order to improve recognition accuracy.

We tested the saliency detector of SalClassNet over saliency benchmarks, where it significantly outperformed existing methods such as SalNet ([19]) and SALICON ([18]) As for evaluating the performance of SalClassNet for visual categorization, we tested it on fine-grained classification tasks over the Stanford Dogs ([107]), the CUB-200-2011 ([8]), and the Oxford Flower 102 ([6]) datasets, showing that explicitly providing visual classifiers with saliency leads to improved performance. As an additional contribution, we release our saliency dataset containing of about 10,000 maps recorded from multiple users when performing visual classification on the 120 Stanford Dogs classes.

We focus our attention, both as building blocks and evaluation baselines, on the SALICON ([18]) and SalNet ([19]) models, thanks to code availability and their fully-convolutional nature.

3.1 SalClassNet: A CNN model for top-down saliency detection

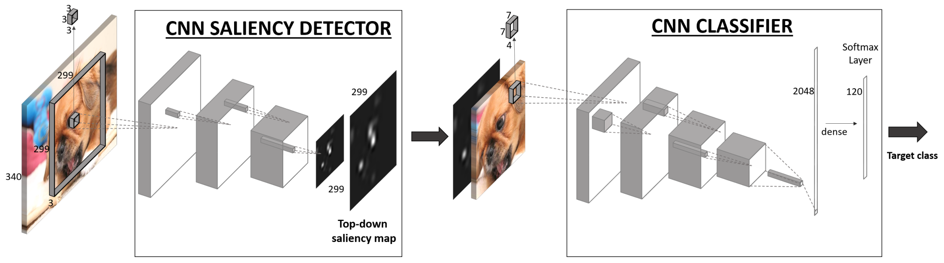


Figure 3.1: *Architecture of the proposed model – SalClassNet– for saliency detection guided by a visual classification task. Input images are processed by a saliency detector, whose output together with input images are fed to a classification network with 4-channel first-layer kernels for processing image color and saliency and providing image classes as output.*

The general architecture of our network is shown in Figure 3.1 and is made up of two cascaded modules: a saliency detector and a visual classifier, which are jointly trained in a multi-loss framework.

3.1.1 Top-down saliency detection network

Although we will discuss the details of the employed saliency dataset and its generation process in Sect. 3.2, it is necessary to introduce some related information at this stage, which is important to understand the overall model.

In the dataset generation protocol, human subjects were explicitly asked to look at images and to guess their visual classes (e.g., dog breeds). Therefore, our experiments aimed to enforce top-down saliency driven by a specific classification task, rather than bottom-up saliency. In other words, instead of emphasizing the location of image regions which are visually interesting per se (which, of course, may include the target object), our visual attention maps focus on the location of features needed for identifying the target classes, ignoring anything else that may be salient but not relevant to the classification task. Hence, our saliency detector has to be able, given an input image, to produce a map of the most salient image locations useful for classification purposes.
To accomplish that, we propose a CNN-based saliency detector composed by thirteen convolutional and five max pooling layers taken from VGG-19 ([108]). The output of the last pooling layer, i.e., 512×10×10 feature maps (for a 3×299×299 input image), is then processed by a 1×1 convolution to compute a saliency score for each "pixel" in the feature maps of the previous layer, producing a single-channel map. Finally, in order to generate the input for the subsequent classification network, the 10×10 saliency maps are upsampled to 299×299 (which is the default input size of the next classification module) through bilinear interpolation.

As for the size of the output maps, it has to be noted that saliency is a primitive mechanism, employed by humans to drive the attention towards objects of interest, which is evoked by coarse visual stimuli ([5]). Thus, increasing the resolution of saliency maps for identifying finer image details from a visual scene is not necessary, beside introducing noisy information potentially affecting negatively the classification performance (indeed, when we increased the saliency map size, the saliency accuracy did not improve). Therefore, in spite of the low spatial resolution of saliency maps, our experiments (see Sect. 3.3) show that the 10×10 feature maps are able to encode the information needed to detect salient areas and to drive a classifier with them.

3.1.2 Saliency-based classification network

Our visual classifier is a convolutional neural network which receives as input a 4-channel RGBS image, combining the RGB image with the corresponding saliency (S) map, and provides as output the corresponding class. The underlying idea is that the network should employ those salient regions (as indicated by the input saliency map S) which are more meaningful for classification purposes.

This network is based on the Inception network ([4]), which comprises sixteen convolutional and one fully connected layer followed by a final softmax layer, with the first-layer convolutional kernels modified to support the 4-channel input. In particular, the 32 3×3×3 kernels in the first layer are converted into 32 4×3×3 kernels, whose weights corresponding to the RGB channels are taken from a pre-trained version of Inception network (see next Sect. 3.3), whereas the new weights, corresponding to the saliency input, are randomly initialized.

Since the model includes a combination of trained weights (the ones from the original Inception) and untrained weights (the ones related to the saliency channel) we set different learning rates in order to speed up the convergence of untrained weights while not destabilizing the already learned ones.

3.1.3 Multi-loss saliency-classification training

The networks described in the previous sections are joined together into a single sequential model and trained using RGB images as input and the corresponding class labels as output. We introduced a batch normalization module between the saliency detector and the classifier, to enforce a zero-mean and unitary–standard-deviation distribution at the classifier's input. During training, we minimize a multi-loss objective function given by a linear combination of cross-entropy classification loss $\mathcal{L}_{\mathcal{C}}$, and saliency detection loss $\mathcal{L}_{\mathcal{S}}$ computed as the mean square error (MSE) of the intermediate saliency detector's output (obtained after the last upsampling layer) with respect to the ground-truth saliency map for the corresponding input image:

$$\mathcal{L}\left(\mathbf{y}, \mathbf{Y}, t, \mathbf{T}\right) = \alpha \; \mathcal{L}_C\left(\mathbf{y}, t\right) + \mathcal{L}_S\left(\mathbf{Y}, \mathbf{T}\right) \tag{3.1}$$

where

$$\mathcal{L}_C(\mathbf{y}, t) = -\sum_{i=1}^{n} \mathbb{I}(i = t) \log(y_i) \tag{3.2}$$

$$\mathcal{L}_S(\mathbf{Y}, \mathbf{T}) = \frac{1}{hw} \sum_{i=1}^{h} \sum_{j=1}^{w} (Y_{ij} - T_{ij})^2 \tag{3.3}$$

where $\mathcal{L}_C$ is the cross-entropy loss computed for the softmax output vector $\mathbf{y}$ and the correct class t, n indicates the number of classes in the dataset, $\mathcal{L}_S$ is the mean square error loss computed on the saliency detector's output map $\mathbf{Y}$ and the ground-truth heatmap $\mathbf{T}$, h and w are the size of the heatmap, and $\mathbb{I}(p)$ is the indicator function, which returns 1 if p is true; bold symbols denote vectors (lower case) and matrices (upper case).

The adopted multi-loss affects the model in several ways. First of all, backpropagating the classification loss to the saliency detector forces it to learn saliency features useful for classification. Secondly, backpropagating the mean square error on the saliency maps ensures that the saliency detector does not degenerate into identifying generic image features and become a convolutional layer as any other.

Figure 3.2 shows two output examples of how saliency changes when using only saliency loss $\mathcal{L}_S$ to train the saliency detector and when driving it by the classification loss $\mathcal{L}_C$: the saliency is shifted from generic scene elements to more discriminative features.

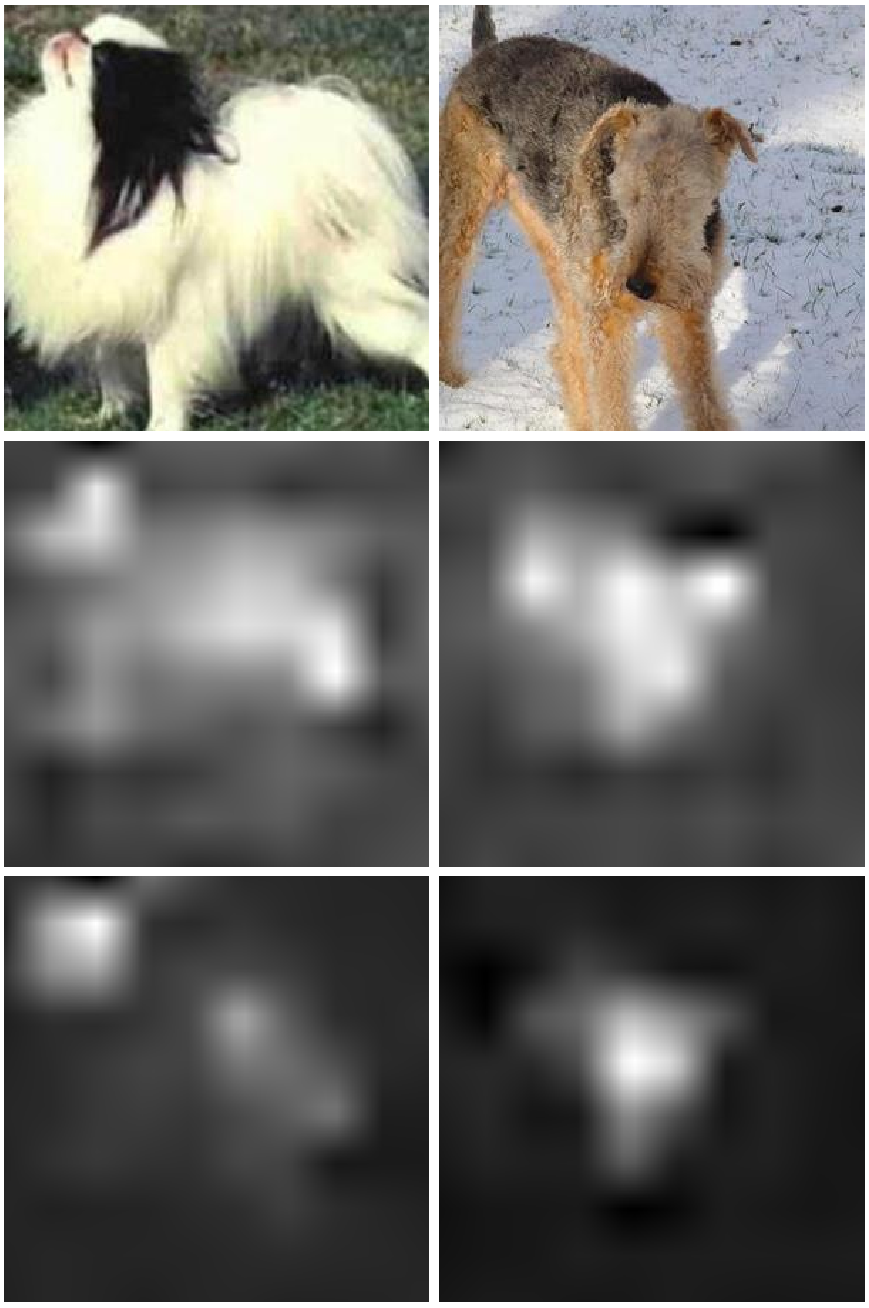


Figure 3.2: **From saliency maps including only sensory information (bottom-up attention processes) to maps integrating task-related information (top-down processes)**. *(Top row) Two example images. (Middle row) Bottom-up saliency maps generated by our CNN-based saliency detector fine-tuned over the Stanford Dog dataset using ground-truth heatmaps. (Bottom row) Shift of saliency guided by the classification task, as resulting from training SalClassNet.*

3.2 Top-down Saliency Dataset

To test our saliency detector, we built a top-down saliency dataset – *SalDogs* – consisting of eye-gaze data recorded from multiple human subjects while observing dog images taken from the Stanford Dogs dataset ([107]), a collection of 20,580 images of dogs from 120 breeds (about 170 images per class). From the whole Stanford Dogs dataset, we used a subset of 9,861 images keeping the original class distribution. The eye-gaze acquisition protocol involved 12 users, who underwent breed-classification training sessions (randomly showing dog images with the related classes), and then were asked to identify the learned breeds from images. To guide top-down visual attention of participants, according to psychology research ([109]), images were blurred with a Gaussian filter whose variance was initially set to 10 and then gradually reduced by 1 each half second until subjects were able to recognize their classes or they were completely de-blurred. Users took, on average, 2.6 seconds to identify dog breeds and 2,763 images were not identified till the end of the de-blurring process. Eye-gaze gaze were recorded through a 60-Hz Tobii T60 eye-tracker. Table 3.1 provides an overview of the *SalDogs* dataset. To the best of our knowledge, this is one of the first publicly-available datasets with saliency maps driven by visual classification tasks, and the first one dealing with a large number of fine-grained object classes. A dataset similar to ours is POET ([110]), which, however, does not deal with fine-grained classification tasks, but with classification at the basic level and with much fewer classes (10 Pascal VOC classes vs 120 in our case). Table 3.2 reports a comparison, in terms of enforced attention mechanism (e.g., tasks accomplished by participants), number of viewers, collected im-

	Our Dataset
Number of images	9,861
Number of classes	120
Avg. number of images per class	82.2
Avg. number of fixation points per image	6.2

Table 3.1: *Information on the generated saliency dataset.*

ages and acquisition devices, between our dataset and recent saliency benchmarking datasets. Finally, to test the generalization capabilities of our saliency detector, we also collected eye gaze data from the same 12 subjects, employing the same data acquisition protocol described above, on: a) bird images (referred in the following as *SalBirds*), using a subset of 400 images taken from CUB-200-2011 dataset ([8]), an image dataset containing 11,788 images from 200 classes representing different bird species; and b) flower images (referred as *SalFlowers*) by selecting 400 images from Oxford Flowers-102 ([6]), which contains over 8,000 images from 102 different flower varieties.

3.3 Performance analysis

The performance analysis focuses on assessing the quality of our model and its comparison to state-of-the-art approaches on two tasks: a) generating task-driven saliency maps from images; b) fine-grained visual recognition task.

Dataset	Task	Viewers	Train	Validation	Test	Tot
SALICON ([37])	Free-viewing	Crowd	10,000	5,000	5,000	20,000
iSUN ([38])	Free-viewing	Crowd	6,000	926	2,000	8,926
MIT300 ([39])	Free-viewing	39	-	-	-	300
CAT2000 ([111])	Free-viewing	24	2,000	-	2,000	4,000
FIGRIM ([112])	Memory	15	-	-	-	2,787
EyeCrowd ([113])	Free-viewing	16	450	-	50	500
OSIE ([114])	Free-viewing	15	500		200	700
PASCAL-S ([115])	Free-viewing	8	-	-	-	850
ImgSal ([116])	Free-viewing	21	-	-	-	235
POET ([110])	Basic classification	28	441	-	5,829	6,270
SalDogs	Fine-grained classification	12	8,005	928	928	

Table 3.2: *Comparison between our dataset and others from the state of the art.*

3.3.1 Datasets

The main benchmarking dataset used for the evaluation of both saliency detection and classification models was *SalDogs* (9,861 images with heatmaps), which was split into training set (80%, 8,005 images – *SalDogs-train*), validation set (10%, 928 images – *SalDogs-val*) and test set (10%, 928 images – *SalDogs-test*).

Specifically for saliency detection, we also employed the POET, *SalBirds* and *SalFlowers* datasets (described in Sect. 3.2) to assess the generalization capabilities of the models trained on *SalDogs*.

For visual classification evaluation, we first carried out a comparison of different models on *SalDogs*, aimed at investigating the contribution of visual saliency to classification. Then, we assessed the generalization capabilities of SalClassNet on the CUB-200-2011 and Oxford Flower 102 fine-grained datasets.

All classification networks (SalClassNet and baseline) were first pre-trained on a de-duped version of ImageNet, obtained by removing from ImageNet the 120 classes present in the Stanford Dogs Dataset. This guarantees fairness between models regardless of pre-training: indeed, since the whole Stanford Dogs is included in ImageNet, publicly-available pre-trained VGG-19 and Inception models would have the advantage of having been trained on images included in *SalDogs-test*.

3.3.2 Training details

The saliency detector in SalClassNet consists of a cascade of convolutional feature extractors initialized from a pre-trained VGG-19, followed by a layer (to train from scratch) which maps each location of the final feature map into a saliency score. An initial pre-training stage was carried out on *OSIE* ([114]), as done also in SALICON. This pre-training employed mini-batch SGD optimization (learning rate: 0.00001, momentum: 0.9, weight decay: 0.0005, batch size: 16) of the MSE loss between the output and target saliency maps; data augmentation was performed by rescaling each image (and the corresponding ground-truth heatmap) to 340 pixels on the short side, while keeping aspect ratio, and randomly extracting five 299×299 crops, plus the corresponding horizontal flips. After this initial pre-training, the resulting model was fine-tuned on *SalDogs-train*: the learning rate was initialized to 0.001 and gradually reduced through the $1/t$ decay rule, i.e., at iteration i it was computed as $l/(1 + 10^{-5} \cdot i)$, with l being the initial learning rate. During this fine-tuning stage, the same data augmentation approach described above and the same values for the other hyperparameters were used.

The saliency-based classifier module of SalClassNet was initially pre-trained as a regular Inception network. Due to the inclusion of Stanford Dogs in ImageNet, we did not employ a publicly-available pre-trained network, and instead trained an Inception architecture from scratch on the de-duped version of ImageNet described in the previous section. We trained the model for 70 epochs, using mini-batch SGD for optimization, with a learning rate schedule going from 0.01 to 0.0001 over the first 53 epochs, weight decay 0.0005 up to the 30^{th} epoch (and 0 afterwards), momentum 0.9 and batch size 32. Data augmentation on the input images was performed as described above. After this pre-training was completed, we modified the first-layer kernels to support RGB color plus saliency input, by adding a dimension with randomly-initialized weights to the relevant kernel tensors, and we fine-tuned the model on *SalDogs-train* for classification, passing as input, each image with the corresponding ground-truth saliency map. Since some weights in the model had already been pre-trained and others had to be trained from scratch, the learning rate was initially set to 0.05 for the untrained parameters, and to 0.001 for the others. We used the same procedures for learning rate decay and data augmentation as in the fine-tuning of the saliency detector, and a batch size of 16.

The final version of the SalClassNet model - which is the one employed in the following experiments - was obtained by concatenating the saliency detector and the saliency-based classifier and fine-tuning it, in an end-to-end fashion, on *SalDogs-train.* Indeed, up to this point, the saliency detector had never been provided with an error signal related to a classification loss, as well as the saliency-based classifier had never been provided with input maps computed by an automated method. Again, the previous procedures for data augmentation and learning rate decay were employed, with a single initial learning rate of 0.001. The α value in Eq. 3.1, weighing the classification loss with respect to the saliency MSE loss, was set to 0.2, since it provided the best accuracy trade-off (see Figure 3.3).

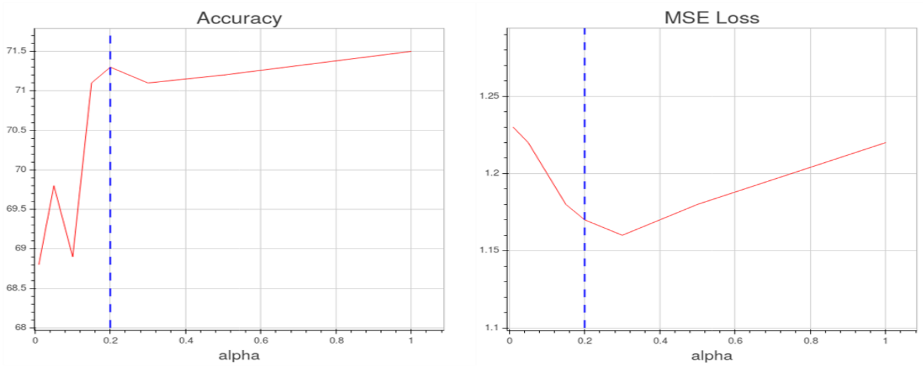


Figure 3.3: *Classification accuracy and MSE w.r.t. α values: 0.2 was chosen as the best trade-off between the two performance metrics.*

During the fine-tuning stages of the individual modules and of the end-to-end model, at the end of each epoch we monitored the classification accuracy and the saliency MSE loss over *SalDogs-val* (evaluating only the central crop of each rescaled image), and stopped training when both had not improved for 10 consecutive epochs: in practice, all models converged in 70-120 epochs. Model selection was performed by choosing the model for which the best relevant accuracy measure (MSE loss for the saliency detector, classification accuracy for the saliency-based classifier and the full SalClassNet model) had been obtained.

3.3.3 Saliency detection performance

To evaluate the capabilities of SalClassNet for saliency detection, we employed the metrics defined by [117] — shuffled area under curve (s-AUC), normalized scanpath saliency (NSS) and correlation coefficient (CC) scores — and compared its performance to those achieved by the SALICON and SalNet models,in their original versions (i.e., as released, pre-trained on the datasets in Table 2.1) and after fine-tuning on *SalDogs-train*.

Table 3.3 reports a quantitative comparison between these approaches over the *SalDogs-test*, POET, *SalBirds* and *SalFlowers* datasets. It is possible to notice that SalClassNet is able to generate more accurate (and generalizes better) top-down saliency maps than existing methods, which suggests that driving the generation of saliency maps with a specific goal does lead to better performance than fine-tuning already-trained models. Figure 3.4 and 3.5 report some output examples of the tested methods on different input images from, respectively, *SalDogs*-test, POET, *SalBirds*, *SalFlowers*. Quantitative and qualitative results show SalClassNet's capabilities to generalize well the top-down visual attention process across different datasets.

	Method	**s-AUC**	**NSS**	**CC**
Dataset	**SalDogs**			
	Human Baseline	0.984	11.195	1
	SalNet	0.720	1.839	0.231
	SALICON	0.805	2.056	0.261
	Fine-tuned SalNet	0.817	4.174	0.432
	Fine-tuned SALICON	0.837	3.899	0.428
	SalClassNet	0.862	4.239	0.461
Dataset	**POET**			
	Human Baseline	0.975	5.189	1
	SalNet	0.646	1.274	0.342
	SALICON	0.723	1.270	0.355
	Fine-tuned SalNet	0.660	1.378	0.300
	Fine-tuned SALICON	0.695	1.669	0.356
	SalClassNet	0.715	1.908	0.387
Dataset	**SalBirds**			
	Human Baseline	0.743	9.323	1
	SalNet	0.642	2.252	0.330
	SALICON	0.680	2.247	0.346
	Fine-tuned SalNet	0.644	3.504	0.403
	Fine-tuned SALICON	0.686	4.252	0.507
	SalClassNet	0.708	4.404	0.529
Dataset	**SalFlowers**			
	Human Baseline	0.975	9.787	1
	SalNet	0.606	1.311	0.1973
	SALICON	0.653	1.081	0.1803
	Fine-tuned SalNet	0.576	0.916	0.136
	Fine-tuned SALICON	0.661	1.599	0.234
	SalClassNet	0.683	1.675	0.245

Table 3.3: *Comparison in terms of shuffled area under curve (s-AUC), normalized scanpath saliency (NSS) and correlation coefficient (CC) between the proposed SalClassNet and the baseline models. For each dataset we report the human baseline, i.e., the scores computed using the ground truth maps.*

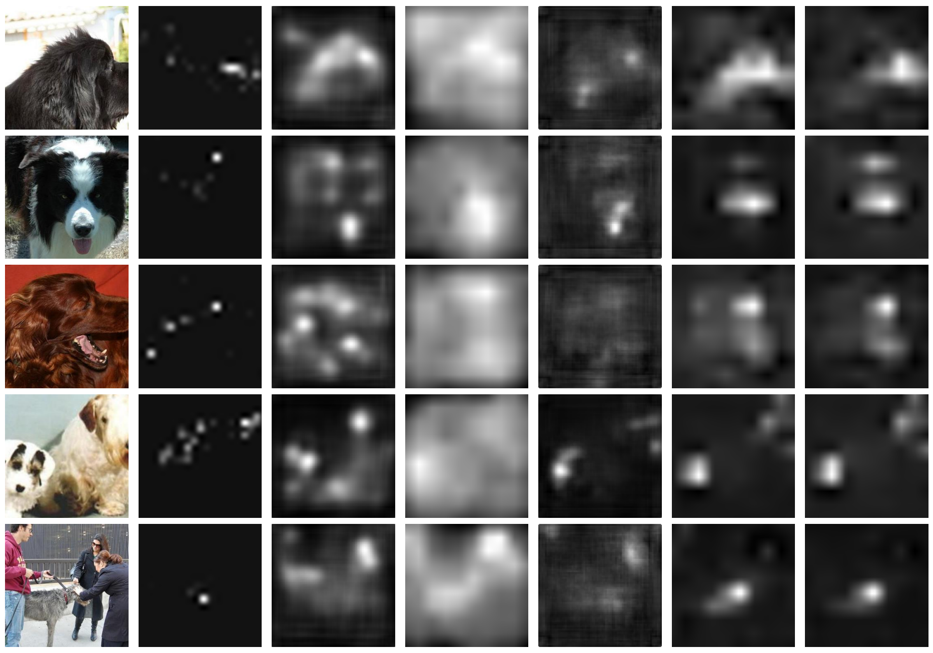


Figure 3.4: Comparison of saliency output maps of different methods. *Each row, from left to right, shows an example image, the corresponding ground-truth saliency map, and the output maps computed, in order, by SalNet and SALICON, as released, and fine-tuned over* SalDogs-train *and the proposed end-to-end SalClassNet model. Beside being able to identify those areas which can be useful for recognition (see first three rows), our method can highlight multiple salient objects (both dogs in the fourth row), or suppress those objects which are not salient for the task (see fifth row).*

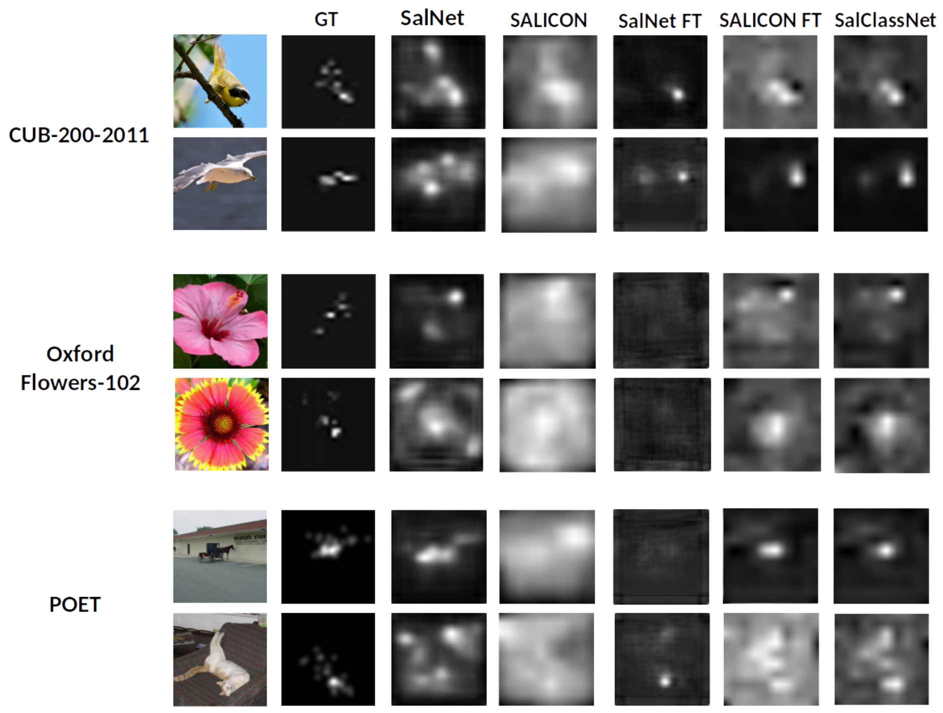


Figure 3.5: *Examples of output saliency maps generated by different methods on* ***CUB-200-2011*** *(first two rows)* ***Oxford Flower 102*** *(third and forth row) and* ***POET*** *(last two rows row) and compared to SALICON-generated saliency maps. Each row, from left to right, shows an example image, the corresponding ground-truth saliency map, and the output maps computed, in order, by SalNet and SALICON, both as released and fine-tuned over* SalDogs-train*, and the proposed end-to-end SalClassNet model. SalClassNet, when compared to SALICON (the second best model in Table 3.3), shows better capabilities to filter out image parts which are salient in general but not necessary for classification.*

3.3.4 Effect of saliency maps on visual classification performance

In this section, we investigate if, and to what extent, explicitly providing saliency maps can contribute to improve classification performance. To this end, we first assessed the performance of VGG-19 and Inception over *SalDogs* when using as input a) only color images (3-channel models) and b) ground-truth saliency maps plus color images (4-channel models). In both cases, as mentioned earlier, we re-trained Inception and VGG-19 from scratch, on the de-duped version of ImageNet and then fine-tuned them on *SalDogs-train*, to force the 4-channel versions to use saliency information coming from the upstream module. Indeed, the publicly-available versions of Inception and VGG had already learned dog breed distributions (trained over 150,000 ImageNet dog images), thus they tended to ignore additional inputs such as saliency. Furthermore, a comparison with Inception and VGG-19 pre-trained on the whole ImageNet would have been unfair also because *SalDogs* contains only about 9,000 images (versus 150,000).

We compared the above methods to our SalClassNet, which automatically generates saliency maps and uses them for classification. Besides the version of SalClassNet described in Sect. **??** (which is also used in all the next experiments), we tested a variant of SalClassNet which employs VGG-19 (suitably modified to account for the saliency input) as classifier: this model is indicated in the results as "SalClassNet (VGG)".

Table 3.4 shows the achieved mean classification accuracies for all the tested methods. It is possible to notice that explicitly providing

Method	**MCA**
VGG (3 channels)	43.4%
VGG (4 channels) + ground truth saliency maps	47.2%
SalClassNet (VGG)	49.0%
Inception (3 channels)	67.1%
Inception (4 channels) + ground truth saliency maps	68.4%
SalClassNet	70.5%

Table 3.4: *Comparison in terms of mean classification accuracy on* SalDogs-test *between the original Inception and VGG models, pre-trained on* ImageNetDD *(ImageNet without the dog image classes) and fine-tuned on* SalDogs-train*, their RGBS variants trained on ground-truth saliency heatmaps and the respective two variants of SalClassNet.*

saliency information (both as ground-truth saliency maps and generated by SalClassNet) to traditional visual classifiers yields improved performance. Indeed, both VGG and Inception suitably extended to make use of saliency information and SalClassNet outperformed the traditional Inception and VGG-19. The lower classification accuracies of the RGBS versions of Inception and VGG (trained with ground truth saliency maps) w.r.t. the SalClassNet variants depend likely by end-to-end training of both saliency and classification networks, which results in extracting and combining, in a more effective way, saliency information with visual cues for the final classification.

Furthermore, SalClassNet showed good generalization capabilities over different datasets, namely, CUB-200-2011 and Oxford Flower 102. In particular, we employed SalClassNet as a feature extractor for a subsequent softmax classifier and compared its performance to those achieved, on the same datasets, by Inception and VGG-19 (fine-tuned on *SalDogs-train* and employed also as feature extractors followed by a softmax classifier). Results are shown in Table 3.5 and confirm our previous claim. The better generalization performance of our method can be explained by a) the fact that the features learned by the classifiers are not strictly dog-specific, but, more likely, belonging to a wider pattern of fine details that can be generally interpreted as significant features (e.g, eyes, ears, mouth, tail, etc.) for classification, thus applicable to a variety of domains; b) SalClassNet, building and improving on the features by Inception, exploits saliency to weigh better the most distinctive features for classification. Hence, although SalClassNet has not been trained on the flower and bird datasets, the generic nature of the learned features and the improved feature filtering gained through saliency led to high accuracy also on those. In order to demonstrate the effectiveness of SalClassNets kernels on different datasets, we computed the features learned by SalClassNet for classification over Stanford Dogs, CUB-200-2011 and Oxford Flowers 102. Table 3.6 shows some of these features, extracted at different SalClassNet depths and visualized by feeding the whole datasets to the network and identifying the image regions which maximally activate the neurons of certain feature maps. It can be seen how meaningful features for dogs turn out to be meaningful for birds and flowers as well.

	CUB-200-2011	**Oxford Flower 102**
Method		
VGG	47.6%	59.2%
Inception	61.8%	77.8%
SalClassNet	**63.2%**	**79.4%**

Table 3.5: *Performance obtained by VGG, Inception and SalClassNet over, respectively,* ***CUB-200-2011*** *and* ***Oxford Flower 102***

3.4 Discussion

In this Chapter, we propose a deep architecture — SalClassNet — which generates top-down saliency maps by conditioning, through the object class supervision, the saliency detection process and, at the same time, exploits such saliency maps for visual classification. Performance analysis, both in terms of saliency detection and classification, showed that SalClassNet identifies regions corresponding to class-discriminative features, hence emulating top-down saliency, unlike most of the existing saliency detection methods which produce bottom-up maps of generic salient visual features. Although we tested our framework using two specific networks for saliency detection and visual classification, its architecture and our software implementation are general and can be used with any fully-convolutional saliency detector or classification network by simply replacing one of the two subnetworks, respectively, before or after the connecting batch normalization module. As further contribution of this work, we built a dataset of saliency maps (by means of eye-gaze tracking experiments

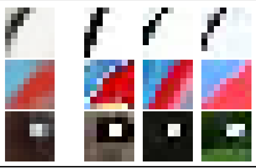
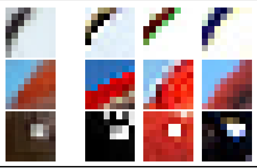
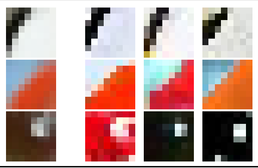
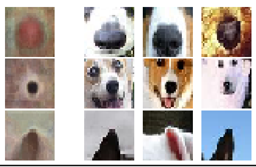
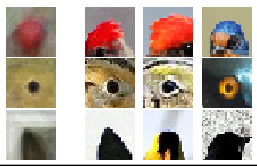
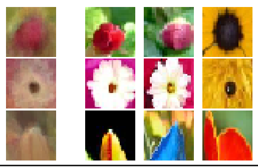
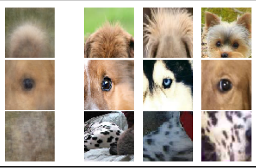
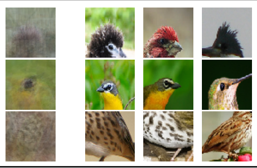
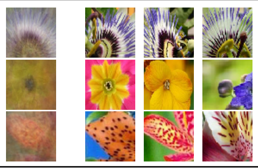
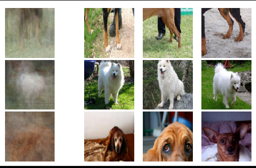
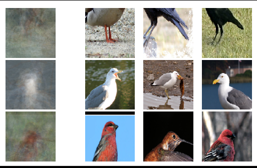
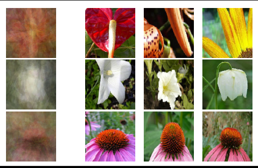
Layer	Stanford Dogs	CUB-200-2011	Oxford Flower 102
1			
5			
11			
16			

Table 3.6: *Examples of features employed by SalClassNet for classification over three different datasets. Each row of images in the tables shows sample which provide high activations for a certain feature map. For each of the tested datasets, we show a 3×4 block of images, where the first column represents the average image computed over the highest 50 activations for that dataset; the last three columns show the three top activations.*

on 12 subjects who were asked to guess dog breeds) for a subset of the Stanford Dog dataset, creating what is, to the best of our knowledge, the first publicly-available top-down saliency dataset driven by a fine-grained visual classification task. We hope that our flexible deep network architecture (all source code is available) together with our eye-gaze dataset will push the research in the direction of emulating human visual processing through a deeper understanding of the higher-level (such as top-down visual attention) processes behind it.

3.5 Publications

- Francesca Murabito, Concetto Spampinato, Simone Palazzo, Daniela Giordano, Konstantin Pogorelov, and Michael Riegler. Top-down saliency detection driven by visual classification. *CVIU*, 2018.

3.6 Released Materials

The *SalDog* dataset, containing of about 10,000 maps recorded from multiple users when performing visual classification on the 120 Stanford Dogs classes, is available at `http://perceive.dieei.unict.it/index-dataset.php?name=Saliency_Dataset`.

CHAPTER

FOUR

GATHERING AND MODELING VISUAL WORLD SEMANTICS FOR FINE-GRAINED IMAGE CLASSIFICATION

In this Chapter we present computational ontologies and show how they can be employed to both represent structured domain knowledge and guide the annotation of a fine-grained visual dataset. To this end, we introduce a *generic visual ontology* and describe its extension in two different domain applications: fruit varieties and cultural heritage buildings. Then, we propose our ontology-based annotation tool and its extension, *CulTO*, specifically designed for the cultural heritage scenario. Finally we present the highly-specialized image dataset *VegImage*,built using our ontology-driven tool and that consists of 3,872 images of 24 fruit varieties and enriched with over 60,000 semantic attributes.

4.1 Ontology-based Modeling of Visual World Semantics

An ontology is a formalism able to provide, for a specific domain, a common machine-processable vocabulary and a formal agreement about the meaning of the used terms, which include relevant concepts, their properties, mutual relations and constraints. Similar to object-oriented programming, basic concepts of a domain correspond to *owl:Class*, whose expressiveness can be enhanced by adding attributes (*owl:DataProperty*) and relations to other *owl:Class* (*owl:ObjectProperty*).

The vocabulary is designed and validated by human users through axioms expressed in a logic language and the concepts and properties can be enriched using natural language descriptions[1]. Thus, an ontology provides an abstract description of objects in a specific domain (e.g., fruit, birds, dogs, cars, etc.), while its instances are well-defined real-world objects (e.g, apple, gull, beagle, pickup, etc.)

Before describing our *generic visual ontology*, let us introduce some terminology. We refer to an *owl:Class* as an *ontology class*, and to a visual class (e.g., fruit, leaf, petiole) as either a *target class* or a *context class*. Target classes represent the main object classes we want to recognize (e.g., *Fruit*): typically, these are visual objects which are spatially well-defined, easily-recognizable and possibly not a constituent part of a larger object (e.g., a fruit rather than a peduncle). Target classes are organized taxonomically and identified in the ontol-

[1]http://www.w3.org/2005/Incubator/mmsem/XGR-image-annotation/

ogy by associating them to SKOS [2] concepts. Instead, context classes are those that either are not classification targets or that are more easily identified in relation to a target class. Their presence is necessary to recognize target instances based on the ontology relations between the two groups.

In order to model generically target/context classes, their relations and visual appearances, we first design a domain-agnostic ontology (referred often in this thesis as *visual ontology*), whose classes can be easily extended to model specific application domains. Figure 4.1 provides a visual representation of the developed ontology. The top-level ontology classes of our *visual ontology* are *PhysicalObject* and *PhysicalProperty*: specifically, *PhysicalObject* is a domain-agnostic class and acts as a base for a hierarchy of target and context classes, which are defined as subclasses of *PhysicalObject* (in a fruit dataset, these could be *Fruit*, *Leaf*, *Peduncle*, etc.); *PhysicalProperty*, instead, is designed to represent domain-specific "attributes" of an object and allows us to provide detailed and structured information on the visual appearance of a *PhysicalObject*; examples of such properties include *Shape*, *Edge* and *Color*. The main relations between classes are *physicalObjectIsPartOf* and its inverse *physicalObjectHasPart*, which link two *PhysicalObject* subclasses related by an *is-part-of* or, more generally, an *is-found-with* relationship. These two transitive properties allow to model all the semantic relations among real world objects needed for visual categorization.

[2]SKOS is a common data model for linking knowledge organization systems and is often used to model taxonomies; SKOS Concepts can be hierarchically organized through the *hasNarrower* or *hasBroader* transitive properties. For more information see *https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html*

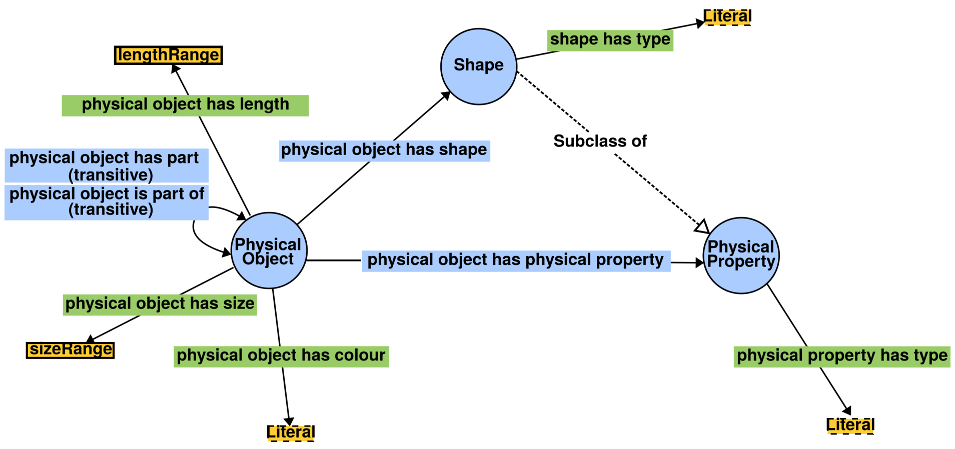

Literal
shape has type
lengthRange
Shape
physical object has length
Subclass of
physical object has shape
physical object has part (transitive)
physical object is part of (transitive)
Physical Object
physical object has physical property
Physical Property
physical object has size
sizeRange
physical object has colour
physical property has type
Literal
Literal

Figure 4.1: *A VOWL representation of our generic visual ontology: PhysicalObject and PhysicalProperty are linked through the generic physicalObjectHasProperty, which can be easily extended to describe any domain-specific attribute (e.g., physicalObjectHasShape).*

4.1.1 Modeling Specific Application Domains

The ontology described in previous section is thought to model generically visual objects and their relations with the real world. One of the main strengths of ontologies is the possibility to easily adapt generic models to very specific domains. Accordingly, we extended our base ontology to model two domains: fruit variety and cultural heritage images.

Fruit ontology. We extend the above *generic visual ontology* by adding entities and relations for visually describing the fruit domain, with the assistance of three expert agronomists, who also supported us in the generation of correct instances of the considered fruit varieties, encoded as instances of the defined ontology which embody all the information we have collected about the classes in our dataset.

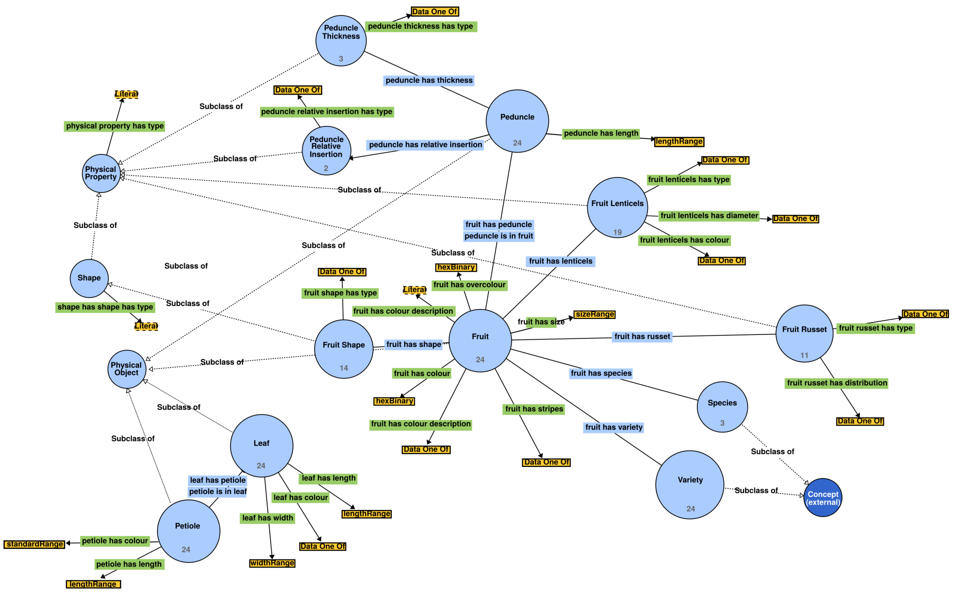
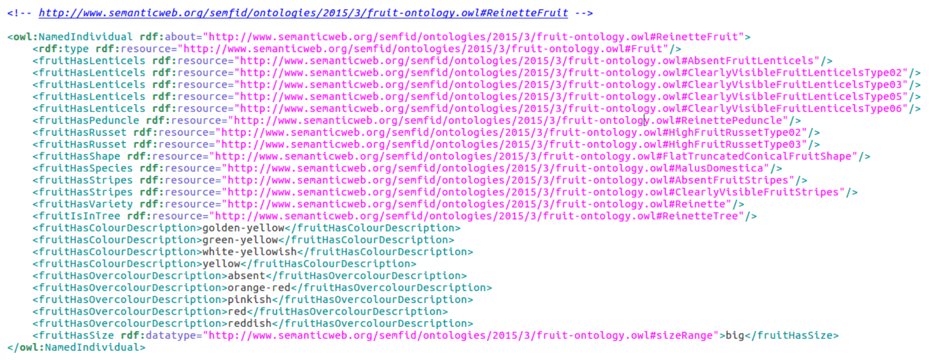


Figure 4.2: *An excerpt of the domain-specific fruit ontology: links to generic classes* PhysicalObject *and* PhysicalProperty *are shown on the left-hand side, while on the right-bottom side are links to the* Species *and* Variety *SKOS concepts (which allow us to define the taxonomy for fruit variety). High-quality image: zoom-in to see more details.*

Figure 4.2 shows an excerpt of the ontology's VOWL (Visual OWL) representation for the fruit classification scenario: *Fruit* and *Peduncle* are defined extending *PhysicalObject* (dotted lines in Figure 4.2), while *FruitRusset*, *FruitShape*, *TreeHabit*, *PeduncleRelativeInsertion* and *PeduncleThickness* are subclasses of *PhysicalProperty* (dotted line in Figure 4.2). The target class (i.e., *Fruit*) is linked to the *Species* and *Variety* classes, which are modeled as SKOS concepts to define species and variety taxonomies. It is important to understand that these ontology classes are not mapped to visual class labels, but refer generically to the categories of objects that we want to classify. The instances of the ontology classes, instead, correspond to the actual visual class labels: for example, the apple variety *Reinette* is encoded as an instance of class *Fruit* (namely, *ReinetteFruit*), while the *ReinetteLeaf* leaf is an instance of ontology class *Leaf* type. The resulting fruit ontology, thus, encodes expert knowledge on visual appearance of different fruit varieties and is used as a basis for our semantic-based classification approach.

Figure 4.3 reports a ontology instance (in XML) describing the *Reinette* apple variety; in particular it shows an instance of *Fruit* class (subclass of the generic *Physical Object*) encoding information about size, stripes, overcolor, color, lenticels, shape and russet for the variety *Reinette*; this information represents the *a priori* knowledge provided by domain experts that are used by our visual-semantic classification approach.

In the instance, all object parts (and their visual properties by exploiting subproperties of the generic *physicalObjectHasPart*) are provided.



```
<!-- http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#ReinetteFruit -->

<owl:NamedIndividual rdf:about="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#ReinetteFruit">
    <rdf:type rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#Fruit"/>
    <fruitHasLenticels rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#AbsentFruitLenticels"/>
    <fruitHasLenticels rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#ClearlyVisibleFruitLenticelsType02"/>
    <fruitHasLenticels rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#ClearlyVisibleFruitLenticelsType03"/>
    <fruitHasLenticels rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#ClearlyVisibleFruitLenticelsType05"/>
    <fruitHasLenticels rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#ClearlyVisibleFruitLenticelsType06"/>
    <fruitHasPeduncle rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#ReinettePeduncle"/>
    <fruitHasRusset rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#HighFruitRussetType02"/>
    <fruitHasRusset rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#HighFruitRussetType03"/>
    <fruitHasShape rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#FlatTruncatedConicalFruitShape"/>
    <fruitHasSpecies rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#MalusDomestica"/>
    <fruitHasStripes rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#AbsentFruitStripes"/>
    <fruitHasStripes rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#ClearlyVisibleFruitStripes"/>
    <fruitHasVariety rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#Reinette"/>
    <fruitIsInTree rdf:resource="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#ReinetteTree"/>
    <fruitHasColourDescription>golden-yellow</fruitHasColourDescription>
    <fruitHasColourDescription>green-yellow</fruitHasColourDescription>
    <fruitHasColourDescription>white-yellowish</fruitHasColourDescription>
    <fruitHasColourDescription>yellow</fruitHasColourDescription>
    <fruitHasOvercolourDescription>absent</fruitHasOvercolourDescription>
    <fruitHasOvercolourDescription>orange-red</fruitHasOvercolourDescription>
    <fruitHasOvercolourDescription>pinkish</fruitHasOvercolourDescription>
    <fruitHasOvercolourDescription>red</fruitHasOvercolourDescription>
    <fruitHasOvercolourDescription>reddish</fruitHasOvercolourDescription>
    <fruitHasSize rdf:datatype="http://www.semanticweb.org/semfid/ontologies/2015/3/fruit-ontology.owl#sizeRange">big</fruitHasSize>
</owl:NamedIndividual>
```

Figure 4.3: *An instance example of* Fruit *class encoding information for Reinette variety.*

Cultural Heritage ontology. We design a specific ontology to model religious historical buildings. The most peculiar elements of these buildings are defined as *Functional Elements*, which are rooms of the construction that absolve a specific function. Crypt, chorus, presbytery, chapel, transept, nave, apse and sacristy are some examples. These structures share the same *Constructive Elements*, such as stairs, horizontal structures, walls and openings, generally found in any other different type of building. Other characteristic structures of churches are *Ancillary Elements*, e.g., altar, baptismal font and pulpit. All these elements, designed in the ontology as subclasses of *PhysicalObject* (which encloses e.g., *Altar*, *BlockAltar*, *Column*, *Capital*, etc.), are characterized by several *PhysicalProperty* (e.g., *Material*). These relationships are shown as arrows in Figure 4.4, containing a partial visual representation of the developed ontology, and specify which class should be considered part of another (e.g., *Capital* is part of *Column*), while blue circles represent classes such as *Capital*, *Shaft*, *ColumnBase* (subclasses of *PhysicalObject*) or *Material* (subclass of *PhysicalProperty*). The developed ontology can be adapted to other building types by creating different subclasses of *PhysicalObject* and *Physical Property* in order to represent the objects belonging to the application domain and their attributes.

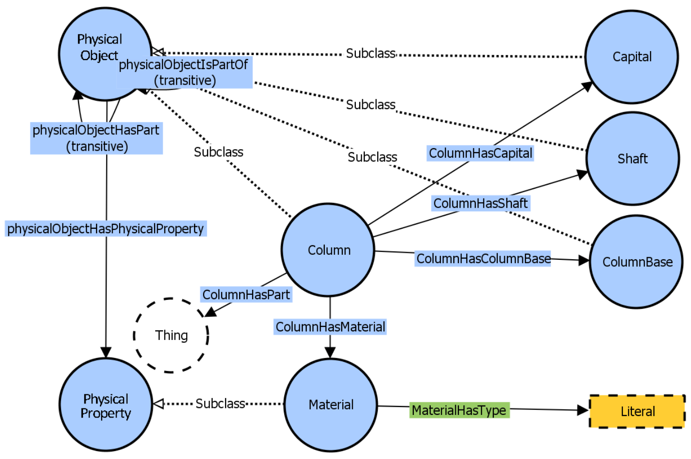


Figure 4.4: *The Visual OWL representation of a subsection of the developed ontology. In particular,* Column, Capital, Shaft *and* ColumnBase *are defined as subclasses of* PhysicalObject *and are linked to each other by relationships in the form of* XHasY*; the column material is in turn defined as a subclass of* PhysicalProperty*. High-quality image: zoom-in to see more details.*

4.2 The Annotation Tool

We exploited our visual ontology to support the image annotation phase, in order to guide and constrain the annotation process, minimizing expert user intervention. More specifically, we developed an ontology-driven annotation tool, which guides and constrains the labeling process within the knowledge base obtained by extending our *visual ontology* (defined in Sect. 4.1 - showing one more time the flexibility of our designed ontology to be extended at will) with the concepts describing the annotation process in a specific application domain. In particular, the link between user annotations and ontology entities is modeled through the *Annotation* class, a subclass of *Sample* class employed to associate sample images, showed to users during the annotation phase, to a *Physical Property.* Since the *Annotation* class is a subclass of *Sample* one, it derives the properties *hasBB* (for "bounding box") and *isInImage*, used to specify the location of an annotated object in an image identifier. Thus, for each new annotation, an *Annotation* instance is automatically created and associated with the corresponding *PhysicalObject* subclass instance (e.g., each annotation of a *Reinette* apple will be implicitly linked to the ontology class instance shown in Figure 4.3); this allows the tool to infer all relevant properties encoded into the ontology (e.g., following the example in Figure 4.3, the *Reinette* apple properties will be: stripes are absent, color is green-yellow, etc.) without the need to specify them manually. Annotator intervention is needed only in cases a property may assume multiple values (e.g., *Russet* for *Reinette* apple), for which, however, the tool displays image examples (whose paths are encoded in the ontology in the *PhysicalProperty*'s *physicalPropertyHasExample* property) to simplify the labeling work for non-experts. Figure 4.5 shows how we extended the main ontology to cope with the visual annotation process.

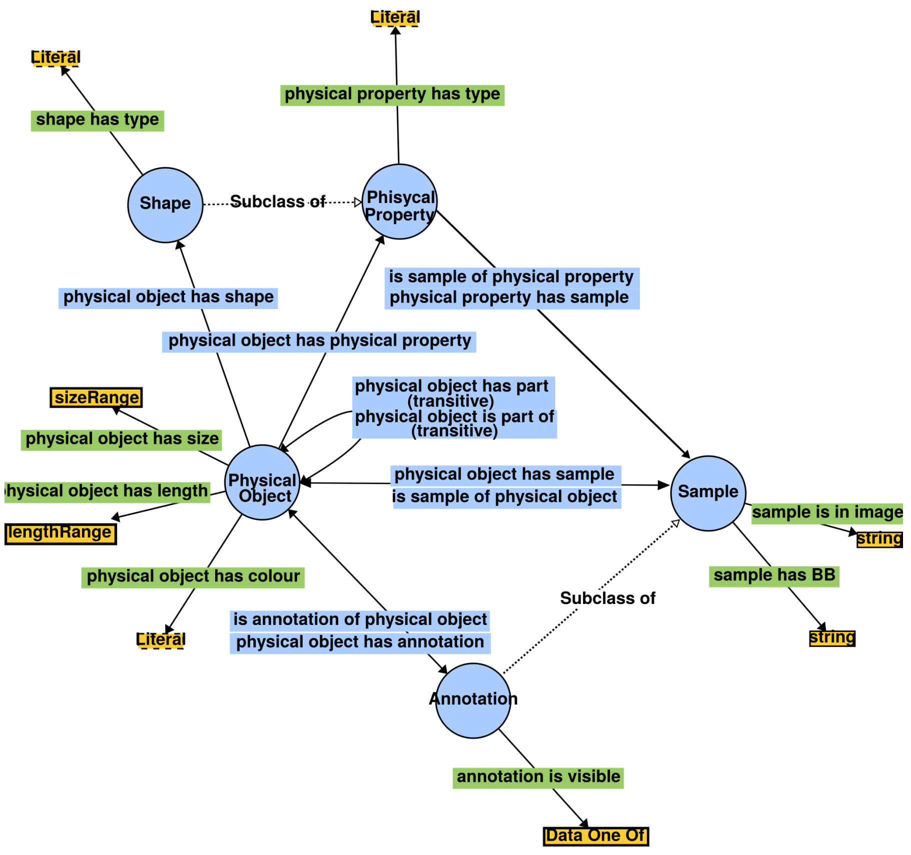

Literal
Literal
shape has type
physical property has type
Shape
Subclass of
Phisycal Property
physical object has shape
is sample of physical property
physical property has sample
physical object has physical property
sizeRange
physical object has part (transitive)
physical object is part of (transitive)
physical object has size
physical object has length
Physical Object
physical object has sample
is sample of physical object
Sample
sample is in image
string
lengthRange
physical object has colour
sample has BB
Subclass of
Literal
is annotation of physical object
physical object has annotation
string
Annotation
annotation is visible
Data One Of

Figure 4.5: *Extension of our visual ontology - described in Figure 4.1 - to support the annotation phase. Linking the annotations to the main ontology (which describes the specific application domain) allows us to infer all the properties of the object being annotated, thus making users save annotation times.*

The main peculiarity of our annotation tool, unlike other existing tools [118, 119, 120], is that part of the label assignment responsibility is moved from the user to the tool itself, through an annotation process divided in two phases, which differ by the degree of expert knowledge required and the amount of annotation work to be carried out.

- The first annotation phase (**expert phase**) consists of freely assigning visual class labels to images. This initial task requires expert knowledge to distinguish between visual classes. In this phase, the expert needs only to annotate *a single* object per image (the image gets the class of the annotated object) in order to provide non-expert users the basis for correctly assigning labels to all other target or context items. For example, in the fruit domain, he/she may just draw one bounding box providing the class for the object therein depicted; in the bird domain, he/she may just annotate the species of a bird or parts thereof.

- Once annotations have been "bootstrapped" by specifying the label for a visual object, the second annotation phase (**non-expert phase**) consists of annotating all other objects present in the image. During this process, we assume that non-expert users are able to identify generic objects in an image corresponding to a visual class (e.g., a bird's beak regardless of bird species a or a generic leaf; in other words, they recognize the appearance of each subclass of *PhysicalObject*).

This annotation step is guided by the tool in four ways:

1. **Annotation of further occurrences of target class:** the tool asks non-expert users to locate all other occurrences of the target class already annotated by experts (who annotate only one object per image).
2. **Sequential annotation of context classes:** let us suppose the expert annotated a *ReinetteFruit* object (a variety of apple); the tool queries the ontology for class instances linked to that object through subproperties of *physicalObjectHasPart* or *physicalObjectIsPartOf*. All the retrieved class instances are valid context classes for the already-annotated object, and for each of them the tool asks the user to locate all occurrences in the image.
3. **Single-valued attribute inference:** single-valued attributes for a given annotation (e.g., "color: red", "shape: round") are encoded in the ontology instance of the corresponding visual class, so the tool automatically infers and associates them to the annotation, without any user intervention.
4. **Multi-valued attribute assignment:** if an attribute may assume multiple values (e.g., the *Reinette* fruit variety can have four types of overcolour property, as shown in Figure 4.3), the tool requires the user select from the possible choices, inferred from the ontology. In order to support the user in the process, the tool also provides visual examples of the possibile values, through the *physicalPropertyHasSample* property (see previous section).

All annotations deriving from points 1, 2 and 3 in the above list are guaranteed to be correct, as long as expert annotations, ontology design, and ontology instances are correct, since all inferences (performed through reasoner [121]) necessarily follow from the initial expert annotations and from known relationships between ontology entities and their attributes. Multi-valued attributes' annotation by non-experts, instead, may need a further validation step by experts.

As an example of the whole annotation procedure for an image, let us take into consideration Figure 4.6 (top), where an expert user annotated only one sample of *ReinetteFruit*. During the non-expert annotation phase, the user is prompted by the tool to annotate all other target class samples and all context class samples (Figure 4.6, bottom), in sequence. In this phase, the non-expert user is not required to specify visual properties for the annotated bounding boxes, as they are directly taken from the underlying ontology.

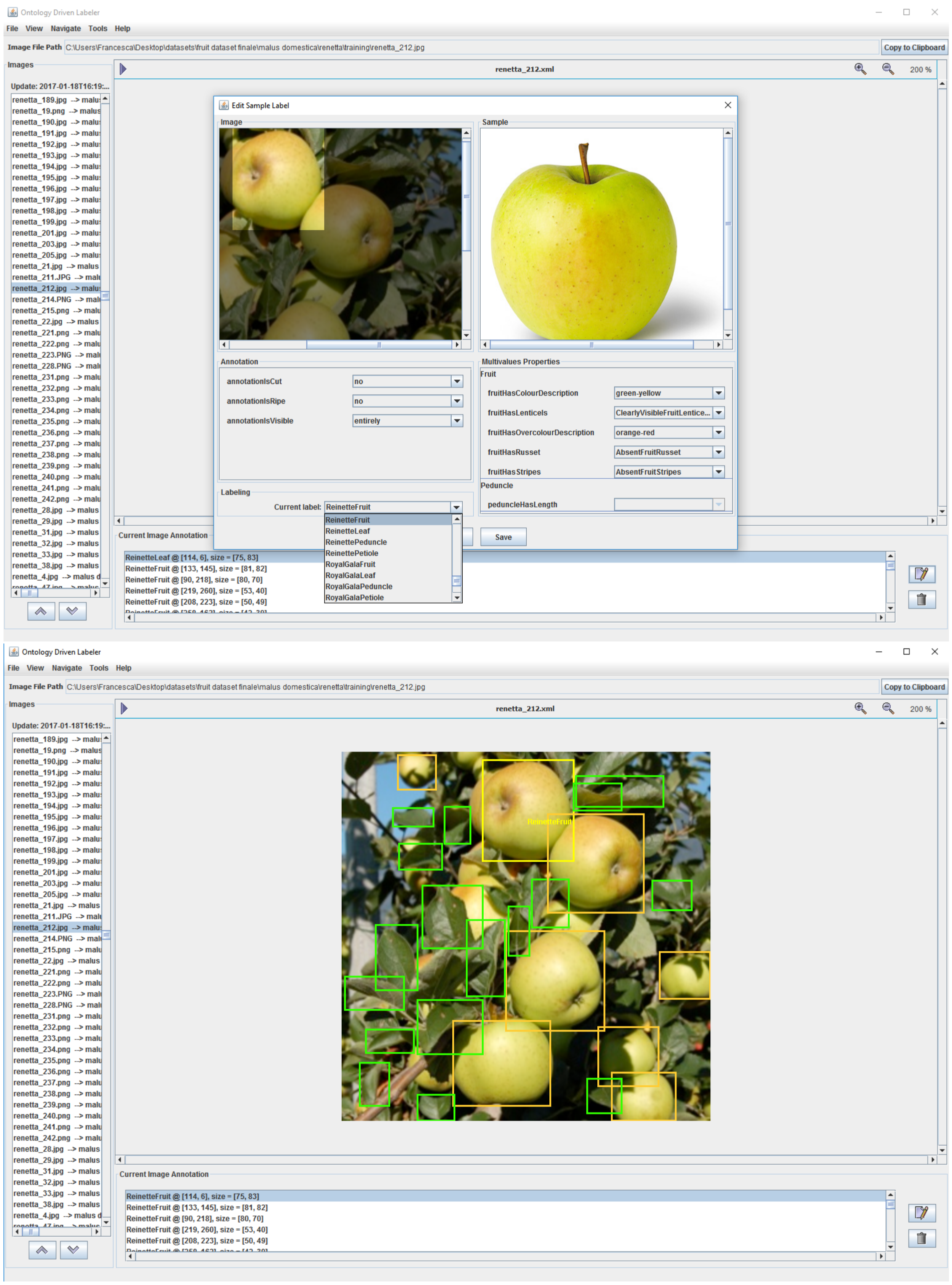


Image File Path C:\Users\Francesca\Desktop\datasets\fruit dataset finale\malus domestica\renetta\training\renetta_212.jpg

renetta_212.xml

Update: 2017-01-18T16:19:...

```
renetta_189.jpg --> malus
renetta_19.png --> malus
renetta_190.jpg --> malus
renetta_191.jpg --> malus
renetta_192.jpg --> malus
renetta_193.jpg --> malus
renetta_194.jpg --> malus
renetta_195.jpg --> malus
renetta_196.jpg --> malus
renetta_197.jpg --> malus
renetta_198.jpg --> malus
renetta_199.jpg --> malus
renetta_201.jpg --> malus
renetta_203.jpg --> malus
renetta_205.jpg --> malus
renetta_21.jpg --> malus
renetta_211.JPG --> malu
renetta_212.jpg --> malus
renetta_214.PNG --> mal
renetta_215.png --> malu
renetta_22.jpg --> malus
renetta_221.png --> malu
renetta_222.png --> malu
renetta_223.PNG --> mal
renetta_228.PNG --> mal
renetta_231.png --> malu
renetta_232.png --> malu
renetta_233.png --> malu
renetta_234.png --> malu
renetta_235.png --> malu
renetta_236.png --> malu
renetta_237.png --> malu
renetta_238.png --> malu
renetta_239.png --> malu
renetta_240.png --> malu
renetta_241.png --> malu
renetta_242.png --> malu
renetta_28.jpg --> malus
renetta_29.jpg --> malus
renetta_31.jpg --> malus
renetta_32.jpg --> malus
renetta_33.jpg --> malus
renetta_38.jpg --> malus
renetta_4.jpg --> malus d
```

Annotation

annotationIsCut	no
annotationIsRipe	no
annotationIsVisible	entirely

Multivalues Properties

Fruit

fruitHasColourDescription	green-yellow
fruitHasLenticels	ClearlyVisibleFruitLentice...
fruitHasOvercolourDescription	orange-red
fruitHasRusset	AbsentFruitRusset
fruitHasStripes	AbsentFruitStripes

Peduncle

peduncleHasLength	

Labeling

Current label: ReinetteFruit

ReinetteFruit
ReinetteLeaf
ReinettePeduncle
ReinettePetiole
RoyalGalaFruit
RoyalGalaLeaf
RoyalGalaPeduncle
RoyalGalaPetiole

Current Image Annotation

```
ReinetteLeaf @ [114, 6], size = [75, 83]
ReinetteFruit @ [133, 145], size = [81, 82]
ReinetteFruit @ [90, 218], size = [80, 70]
ReinetteFruit @ [219, 260], size = [53, 40]
ReinetteFruit @ [208, 223], size = [50, 49]
```

Current Image Annotation

```
ReinetteFruit @ [114, 6], size = [75, 83]
ReinetteFruit @ [133, 145], size = [81, 82]
ReinetteFruit @ [90, 218], size = [80, 70]
ReinetteFruit @ [219, 260], size = [53, 40]
ReinetteFruit @ [208, 223], size = [50, 49]
```

Figure 4.6: **Top**: *bounding box annotation of a target object. Experts have to annotate only one target object per image. Experts have to annotate only one bounding box and specify for it the class of the target object (in the example, the user annotated one example of Reinette Apple). Once this annotation has been done, non experts users have to annotate of other instances of the target object (in the example other instances of Reinette Apple in the image) as well as context objects (e.g., leaves, petioles, etc.) -* **Bottom**. *All the non-expert provided bounding box labels are inferred automatically from expert annotation (top image) through ontology.*

4.2.1 CulTO

CulTO is an extension of the tool presented in the previous section specifically designed for the annotation of images depicting cultural heritage buildings. It provides means to draw a polygon on a specific object and to link this annotation to an ontology instance in order to derive all the instance properties (e.g., the kind of *Material* or their *Shape*), already defined in the ontology itself, and the precise object class, representing the type of the annotated part (e.g., *Altar*, *Column*, etc.). Moreover, the user may add a text description of the current annotation (e.g., the presumed altar dedication) and select some other properties predefined in the ontology for each visual annotation (e.g., the object visibility) as shown in Figure 4.7, top. Furthermore, the tool enable the user to tag the position where an object is found, enabling a successive post-processing stage. Finally, in order to annotate unknown objects, it is possible to insert additional instances selecting the class whose the object belongs to (see Figure 4.7, bottom), allowing the tool to augment dynamically the knowledge about the current application domain.

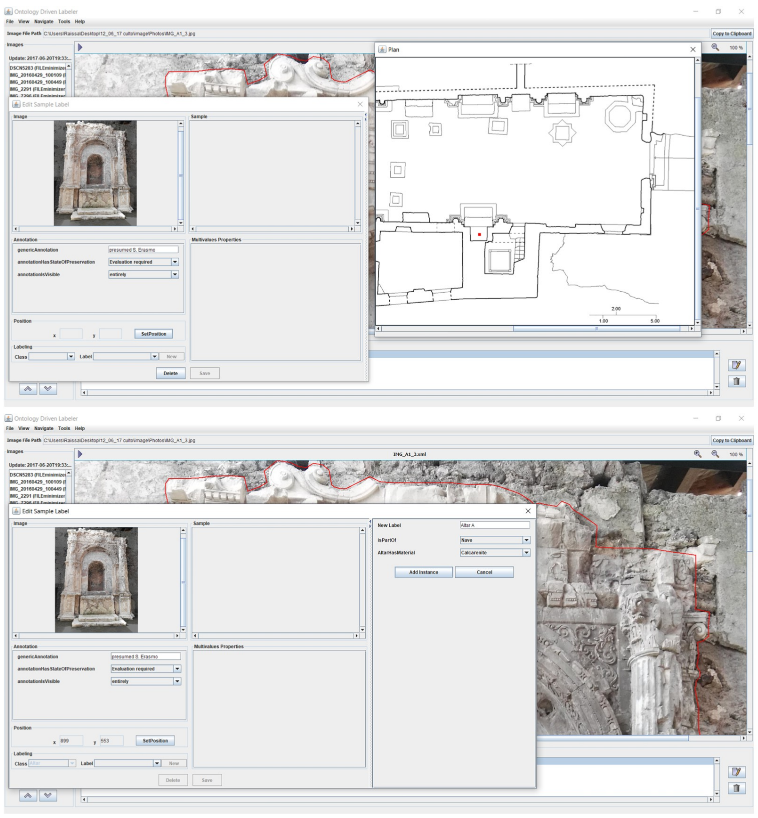


Image File Path C:\Users\Raissa\Desktop\12_06_17 culto\image\Photos\IMG_A1_3.jpg

Update: 2017-06-20T19:33:...

DSCN5283 (FILEminimizer
IMG_20160429_100109 (F
IMG_20160429_100449 (F
IMG_2291 (FILEminimizer

IMG_A1_3.xml

genericAnnotation	presumed S. Erasmo
annotationHasStateOfPreservation	Evaluation required
annotationIsVisible	entirely

x 899 y 553

Class Altar Label

New Label	Altar A
isPartOf	Nave
AltarHasMaterial	Calcarenite

Figure 4.7: **Top**: *The user is prompted with a dialog box where he/she is able to tag the position with a red dot.* **Bottom**: *Once the user has selected the class Altar and clicked the "New" button, the dialog box allows the user to add a new instance in the current ontology and specify all its attributes (e.g., the altar material).*

4.3 VegImage - A Semantic Fruit Image Dataset

This section describes the fruit image dataset, *VegImage* highlighting the role that ontologies had in the annotation process. Indeed, our claim is that there are cases when relying only on low and middle level visual features are not enough to perform visual classification. This especially holds for small and highly-specialized datasets. To demonstrate this, we built a benchmarking dataset- using the tool described in the previous Section - *VegImage*: a collection of 3,872 images, semantically-enriched with information about appearance and context, of three common fruit species, namely, *malus domestica* (apple), *prunus avium* (cherry) and *pyrus communis* (pear). For each fruit species, several fruit varieties were included: 10 for *malus domestica*, 7 for *prunus avium* and 7 for *pyrus communis*. Together with fruit images, we also generated over 65,000 bounding boxes (depicting the different varieties of fruits, leaves, peduncles, etc.) and a large knowledge base (over 1,000,000 OWL triples) containing high-level knowledge on context objects and attributes for the considered fruit varieties.

Dataset collection. The fruit variety images were mainly downloaded from Google Images, Flickr, ImageNet and Yahoo Images. Given the specificity of the task, for some varieties it was extremely complex to collect a significant number of samples and for this reason, we also downloaded YouTube documentary videos, from which we manually selected key frames to avoid near duplicates in the dataset. For each of the 24 fruit varieties (depicted in Figure 4.8, about 500 images were manually selected to be included in the dataset. Near duplicates, low-quality images or images depicting multiple fruit varieties or people as main subjects were filtered out. After this screening, three expert Agronomists manually checked thoroughly all the resulting images. Thus, we collected few hundreds images for each fruit variety.

Dataset annotation. We performed a two-stage annotation phase using an annotation tool (described in the following) able to exploits ontologies to support the annotation phase: a) **Image labeling**: in this step, the three agronomists annotated each image with a label decided through consensus among them; b) **Bounding box annotation**: ten non-expert users were asked to draw bounding boxes (a distribution over fruit varieties is given in Table 4.1) for objects of both target (*Fruit*) and context classes (*Peduncle*, *Leaf* and *Petiole*), and to disambiguate multi-valued attributes defined in the Fruit Ontology (e.g., russet for *Reinette* apple), which were finally double-checked by the experts, being the only kind of annotations which could be subject to errors.

Malus Domestica						
	#img	*#bounding_boxes*				
Variety		Fruit	Peduncle	Leaf	Petiole	Total
Ambrosia	117	1878	1,218	362	144	3,602
Braeburn	83	405	847	118	94	1,464
Cameo	70	410	210	163	30	813
Fuji	181	2,216	837	230	111	3,394
Golden Delicious	324	2,536	1,770	619	257	5,182
Granny Smith	360	825	434	371	93	1,723
Pink Lady	239	1,145	1,004	283	215	2,647
Renetta	242	1,117	719	231	111	2,178
Royal Gala	150	973	538	266	119	1,896
Stark Delicious	411	1,661	1,406	479	142	3,688
Total	2,177	13,166	8,983	3,122	1,316	26,587
Prunus Avium						
	#img	*#bounding_boxes*				
Variety		Fruit	Peduncle	Leaf	Petiole	Total
Bing	67	1,483	551	532	111	2,677
Black Tartarian	45	542	264	329	123	1,258
Burlat	71	1,099	690	564	215	2,568
Ferrovia	77	2,930	388	1,838	151	5,307
Lapins	87	1,678	430	751	178	3,037
Rainer	140	1,854	776	836	383	3,849
Stella	31	595	221	251	106	1,173
Total	518	10,181	3,320	5,101	1,267	19,869
Pyrus Communis						
	#img	*#bounding_boxes*				
Variety		Fruit	Peduncle	Leaf	Petiole	Total
Abate	286	1,686	1,379	718	510	4,293
Anjou	191	1,031	978	486	513	3,008
Conference	181	1,009	747	523	307	2,586
Coscia	166	898	1,087	479	499	2,963
Doyenne du Comice	77	459	278	274	175	1,186
Kaiser	111	520	275	276	99	1,170
Williams	165	2,057	1323	911	526	4,817
Total	1,177	7,760	6,067	3,667	2,629	20,023
Dataset	3,872	31,007	18,370	11,890	5,212	66,479

Table 4.1: *Number of images and bounding boxes for each fruit variety. In total, the Fruit Image Dataset contains 3,872 images and 66,479 bounding boxes of fruits, leaves, etc.*

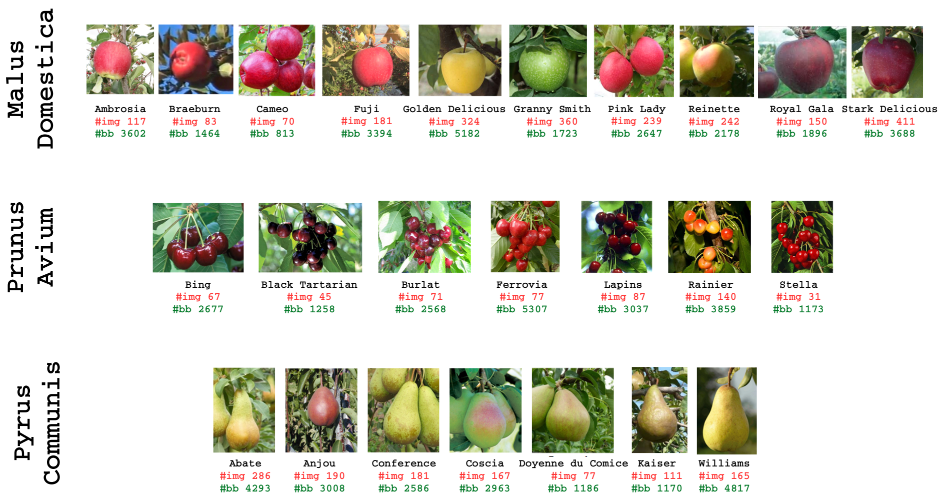

Malus Domestica
Ambrosia #img 117 #bb 3602
Braeburn #img 83 #bb 1464
Cameo #img 70 #bb 813
Fuji #img 181 #bb 3394
Golden Delicious #img 324 #bb 5182
Granny Smith #img 360 #bb 1723
Pink Lady #img 239 #bb 2647
Reinette #img 242 #bb 2178
Royal Gala #img 150 #bb 1896
Stark Delicious #img 411 #bb 3688
Prunus Avium
Bing #img 67 #bb 2677
Black Tartarian #img 45 #bb 1258
Burlat #img 71 #bb 2568
Ferrovia #img 77 #bb 5307
Lapins #img 87 #bb 3037
Rainier #img 140 #bb 3859
Stella #img 31 #bb 1173
Pyrus Communis
Abate #img 286 #bb 4293
Anjou #img 190 #bb 3008
Conference #img 181 #bb 2586
Coscia #img 167 #bb 2963
Doyenne du Comice #img 77 #bb 1186
Kaiser #img 111 #bb 1170
Williams #img 165 #bb 4817

Figure 4.8: *Example images from the* VegImage *dataset. Numbers in red are number of images per class while in green the number of bounding boxes.*

Comparison to existing datasets for visual categorization. Table 4.2 compares the Fruit Image dataset with three popular benchmarking datasets for fine-grained image classification: Oxford-IIIT Pet [7], Oxford Flower 102 [6] and Caltech-UCSD Birds (CUB-200-2011) [8]. Although the Fruit Image dataset contains less categories and images than Oxford-IIIT Pet, Oxford Flower 102 and CUB-200-2011, it is more complete in terms of objects per image, parts per object and attributes per object, besides being the only one enriched with a large structured knowledge base. Furthermore, the number of objects per image $O/I = 8.0$ in our dataset greatly outnumbers existing fine-grained datasets. The number of parts per object $P/0$ is sensibly greater in CUB-200-2011 (12.0) than in ours (1.14) since birds are more structured objects than fruits. However, our parts per object refer more to context objects while in CUB-200-2011 are parts of the objects themselves. Also the number of attributes per object A/O in CUB-200-2011 is almost three times than ours since it is directly connected to the object parts.

	#C	**#I**	**I/C**	**O/I**	**P/O**	**A/O**
Fine-grained datasets						
Oxford-IIIT Pet [7]	37	7,349	198.6	1.0	-	-
Oxford Flower 102 [6]	102	8,189	80.3	1.0	-	-
CUB-200-2011 [8]	200	11,788	58.9	1.0	12.0	31.5
Fruit Image	24	3,872	161.3	8.0	1.14	11.0

Table 4.2: *Comparison between popular visual datasets and our dataset. Key:* #C*: number of classes;* #I*: number of images;* I/C*: average number of images per class;* O/I*: average number of objects per image;* P/O*: average number of parts per object;* A/O*: average number of attributes per object. For our dataset, the* O/I *value refers to the number of target objects, whereas the* P/O *value counts context objects as object parts; object attributes are the OWL triples.*

To better understand the benefits that our tool and computational ontologies provided to the annotation phase we assessed: 1) workload shift from experts to non-experts and 2) non-expert annotation time. Domain experts manually annotated 3,872 fruit images, while 66,479 bounding boxes, and their attributes, were provided by ten non-expert users. Bounding box attributes were inferred automatically by the OWL reasoner (through deductive inference encoded in Protege[3]: the tool we used for ontology development) after the corresponding bounding box class (e.g. *Leaf*) and variety (e.g., *Cameo*) were specified. The annotations of 3,872 fruit images by the three experts took about 9 days (average of about 1.3 hours per day per expert) for a total of 35 (expert) worker hours, while the annotation of 66,479 bounding boxes took about 12.5 days (average of 2.5 hours per day per annotator). In total, annotating the whole image dataset took 349 worker hours: 314 (about 90% of the total) hours provided by non-experts and the remaining 35 hours by experts.

The average annotation time per bounding box for non-experts was 17.1 seconds, which is impressive given that the *VegImage* deals with a specific and complex application domain, and considering, for instance, that in COCO [122] (a large-scale dataset for basic image classification) the annotators spent, on average, about 80 seconds per bounding box.

As a final note, our ontology-based tool allows to tackle the issue recently reported in [123], i.e., high-quality annotations on domain-specific applications should be performed, if not by experts, at least by citizen scientists, since unskilled workers perform extremely bad. While this may hold for "traditional" annotation approaches, encoding and incorporating domain knowledge in a tool able to constrain the labeling process demonstrated to be a valid alternative, allowing non-expert annotators to provide high-quality annotations while saving significantly expensive (expert) resources.

[3]http://protege.stanford.edu/

4.4 Discussion

In this Chapter we first introduced computational ontologies as a way to model high-level visual information about objects, their context and appearance and showed their generalization capabilities in modeling two different application domains: fruit variety and cultural heritage. We then presented a knowledge-driven annotation tool which exploits specialized domain knowledge to generate semantic annotations, significantly reducing the efforts of domain experts, for classification problems. The tool was used by three expert agronomists to provide high-level and coarse annotations and by ten non-expert users who provided fine-grained annotations without any knowledge on the application domain. The resulting *VegImage* dataset contains 3,872 images, over than 60,000 bounding boxes, and over than 1,000,000 OWL triples, representing, to the best of our knowledge, one of the most comprehensive resources for fine-grained classification and one the most exhaustive knowledge bases in computer vision. In the next Chapter, we will describe a machine learning classifier leveraging the built knowledge base to perform fine-grained recognition.

4.5 Publications

- Francesca Murabito, Simone Palazzo, Concetto Spampinato, and Daniela Giordano. Generating knowledge-enriched image annotations for fine-grained visual classification. In *ICIAP*, 2017.

- R Garozzo, F Murabito, C Santagati, C Pino, and C Spampinato. Culto: an ontology-based annotation tool for data curation in cultural heritage. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2017.

4.6 Released Materials

Our *VegImage* dataset, images and high-level annotation, is available for download at `http://perceive.dieei.unict.it/index-dataset.php?name=Fruits_Dataset`.

CHAPTER

FIVE

VISUAL SEMANTICS FOR FINE-GRAINED VISUAL CLASSIFICATION

This Chapter tackles the problem of building machine learning models exploiting jointly purely-visual features and high-level structured knowledge described through a formal ontology. We then present the results achieved on the *VegImage* dataset, introduced in the previous Chapter, and show how our classifier, compared to the state-of-the-art methods, yields higher accuracy, thus demonstrating that the exploitation of the semantic information can indeed help support image recognition.

5.1 Semantic-based Image Classification

As mentioned before our classifier is made of two blocks: a) a knowledge-based classifier, and b) a standard CNN classifier relying only on visual features. Our main focus is on describing the former, as it represents the novel aspect — classification-wise — of this model; we see later how the predictions from the two classifiers are merged.

Knowledge-based classifier. The underlying idea of our semantic-based classifier consists of building a Bayesian Network graph as the one depicted in Figure 5.1, where each node represents a *PhysicalObject* or a *PhysicalProperty* defined in our ontology and each edge corresponds to an interconnection between nodes in terms of *being-part-of-this-object* or *being-identified-by-this-property*, and performing recognition through inferences over the graph. We opt for using graphs since they inherently allow to encode ontology relations, such as co-occurrence (through the graph definition) and mutual exclusion by properly setting the conditional probabilities among graph nodes, representing ontology entities. In particular, given an ontology instance $\mathcal{O}$, we built a graph $G^{\mathcal{O}}(N, E)$, with $N = N_1 \cup N_2$ (where N_1 describes the set of target and context classes and N_2 the set of visual properties)nodes and E edges. For example, in the fruit classification task, we have $N = 12$ (corresponding to the number of circles depicted in Figure 5.1) given by $N_1 = 4$ main visual classes (*Fruit*, *Leaf*, *Petiole*, *Peduncle*) and $N_2 = 8$ class properties defined in the ontology (e.g., *FruitStripes*, *FruitColor*, *LeafShape*, etc.); we leave out those properties which cannot be visually identified since they encode quantitative information about the target class (such as *FruitSize*).

Each edge is identified by a matrix containing the conditional probabilities between object classes and visual properties; in particular, being P the number of possible values that can be assumed by a property and V the number of object classes (in the case of fruit classification V is set to 24), it is possible to define a matrix $[C]_{P\times V}$ where each element c_{ij} represents the conditional probability

$$c_{ij} = p(v = v_j|p = p_i) \tag{5.1}$$

i.e., the probability of the variety v_j given the property value p_i. In order to compute these conditional probabilities we exploit the expert knowledge encoded in the domain ontology; specifically, being f_i the absolute frequency of the property value p_i, i.e., the number of its occurrences in the ontology instances (e.g., considering the property *FruitColor*, the absolute frequency of property value *Yellow* is computed as the number of varieties whose fruit color is *Yellow*), we set the conditional probability as follows:

$$c_{ij} = \begin{cases} \frac{1}{f_i}, & \textit{if the variety } v_j \textit{ exhibits the property } p_i. \\ 0, & \textit{otherwise.} \end{cases} \tag{5.2}$$

thus, assigning low conditional probabilities to those properties' values which cannot be considered distinctive of a specific object class. Therefore, the structure of this Bayesian Network graph depends only on the knowledge (classes, relations between classes and their instances) contained in the Fruit ontology; given this graph, a new test image is classified by performing inferences based on the structure itself (in terms of nodes, edges and conditional probabilities) and the initial evidences computed for each node, which are then propagated throughout the graph (see Figure 5.1 for an example).

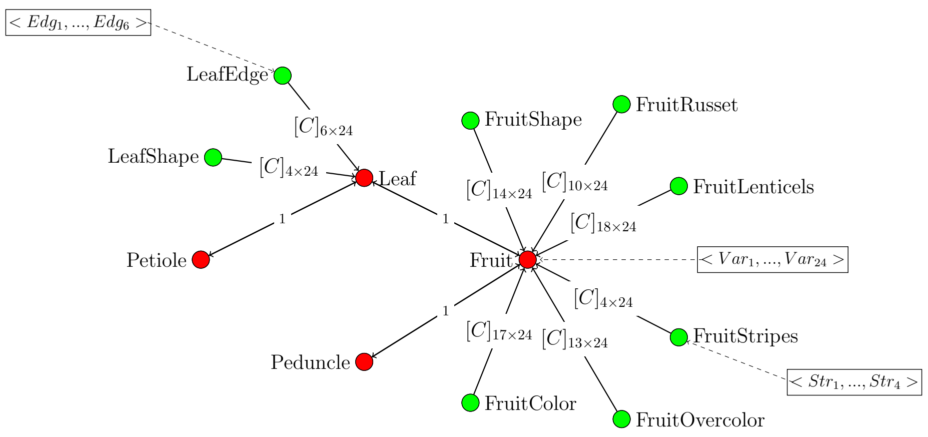


Figure 5.1: *An example of the Bayesian Network graph for the fruit classification task. Red nodes are visual classes while green ones correspond to visual properties. For simplicity, initial evidences of only one visual class (*Fruit*) and two visual properties (*FruitStripes *and* LeafEdge*) are shown.*

Formally, our method consists of the following steps:

(a) **Compute visual evidences for test images.** For each new test image I, we suppose an uniform distribution of the visual class labels; hence, for each visual class node (i.e., the red ones in Figure 5.1), we set the initial evidences to

$$e =< \frac{1}{V}, ..., \frac{1}{V} > \tag{5.3}$$

i.e., an array of $\frac{1}{V}$ with dimension V. For visual properties (i.e., the green nodes in Figure 5.1) we apply a three-phase approach: 1) first, we employed the object detectors to localize objects belonging to a visual class (*Fruit*, *Leaf*, *Peduncle*, etc.); 2) afterwards, we apply a non-maximum suppression, thus leaving, for each visual class, only one detected object, corresponding to the detection with higher likelihood for that class; 3) finally, on the filtered objects, we run CNN-based pretrained classifiers and use its features maps to detect classes' attributes (e.g., color, shape, etc.; all properties are specified in the ontology).

(b) **Perform Belief Propagation** Since each image is grounded to a directed acyclic graph, we could use Conditional Random Field to assess the marginal probabilities of unobserved variables as in semantic relation graphs [12] or in scene graphs [64]. This, however, besides requiring large annotated data, would affect its generalization capabilities since it would need a new training of CRF for each scenario, while our approach needs only to train the detectors (as also in [12, 64]). Thus, we compute class red labels' probabilities by performing inferences over the graph.

In particular, each graph can be seen as a Bayesian Network with N random variables x_i defining the joint probability function

$$p(x_1, x_2, ...x_n) = \prod_{i=1}^{N} p(x_1)|par(x_i)) \tag{5.4}$$

We compute marginal probability or belief ($p(x_i)$ or b_i) for the graph node corresponding to the target class; the result represents the distribution of visual class labels' beliefs.

Marginal probabilities are computed through Belief Propagation (BP) [124], which has been successfully employed to perform scene understanding based on the use of priors [125].

Joint visual-semantic classifier. Alongside our knowledge- based approach, we train a purely-visual classifier. Indeed, in spite of the above-mentioned limitations of current deep models for fine-grained recognition, they can still provide necessary information complementary to the one harnessed from encoded expert knowledge. Therefore, we compute the visual probability distribution P_I^v as the output of a convolutional neural networks softmax layer.

The final class prediction for test image I is carried out by computing a prediction vector P_I as the combination of the semantic probability distribution P_I^s and the visual probability distribution P_I^v. In particular, in order to combine the two distributions, we opt for their weighted sum; specifically the prediction vector is computed as

$$P_I = O_V + \alpha * P_I^s \tag{5.5}$$

where P_I^v represents the visual classifier distribution, P_I^s the semantic classifier one and α is parameter which scale the probability marginals (or beliefs) before their combination with the CNN softmax output. The resulting P_I may not strictly be a probability distribution (as the sum of the elements can be greater than 1), but we are only interested in retrieving the maximum value, as that will be the final prediction for the class c assigned the input image:

$$c = \arg\max_i P_I \tag{5.6}$$

Figure 5.2 shows an example for how the whole classification process works. In particular, for each test image I, we run an object detector to identify, for each visual class, (i.e., *Fruit*, *Leaf*, *Peduncle* and *Petiole*) the location with higher probability. Then, on the selected locations, visual properties' detectors are applied to compute the initial evidence (likelihood) for each property; once these evidences are obtained, they are propagated throughout the Bayesian Network graph by means of the Belief Propagation and the resulting marginals' probabilities distribution is combined with its counterpart obtained from the visual-only classifier in order to estimate the image class for the test image.

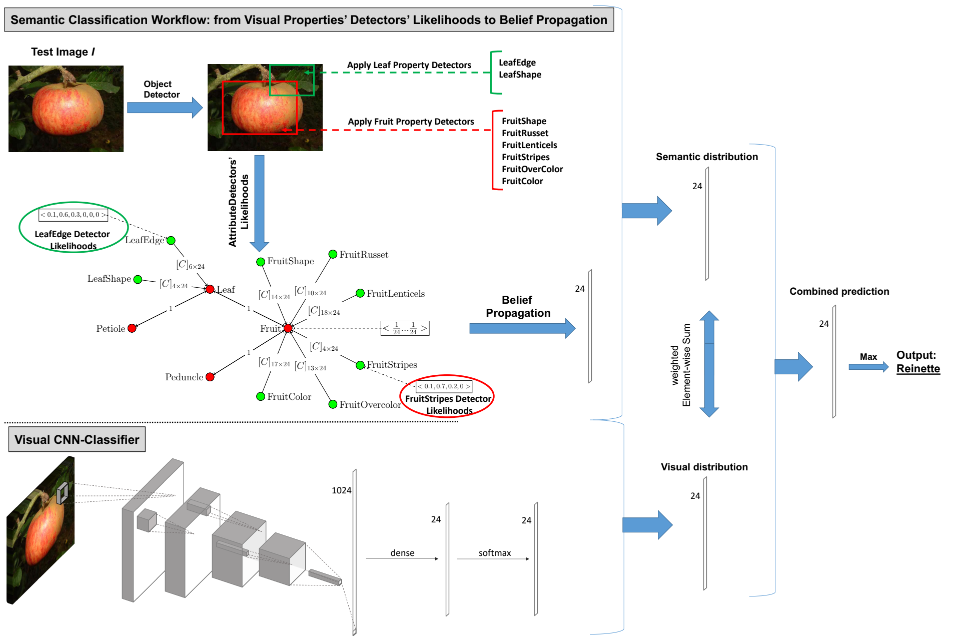


Figure 5.2: *Example of the proposed semantic-based classifier.* **Top***: semantic classification workflow for test image I. In the example, we run a Fast R-CNN detector and apply non-maximum suppression independently, thus obtaining two locations, one for the visual class* Fruit *and another for the visual class* Leaf*. Then, on each location, the corresponding properties detectors are applied in order to compute the properties likelihoods used as initial evidences in the Bayesian Network graph. The application of the Belief Propagation provides the marginal probabilities distribution over the labels, resulting in a 24-dimensional vector.* **Bottom***: Finally, we compute the softmax output of the visual-only CNN classifier applied on the entire image and combine the visual-based class distribution with the semantic one in order to predict the correct classification output. For clarity, initial evidences of only one visual class (*Fruit*) and two visual properties (*FruitStripes *and* LeafEdge*) are shown.*

5.2 Performance Analysis

This section reports the experimental evaluation we carried out to demonstrate the advantages that structured semantics bring to fine-grained image classification providing the results achieved by our method on the *VegImage* and comparing it to the state of the art.

As shown in Sect. 5.1, our knowledge-based image classification approach relies on CNN-based object detector to detect instances of target and context classes and CNN-based classifier for classification of object visual properties. As object detector, we employed Fast R-CNN detection model [126]. In particular, we used the configuration based on the 8 pre-trained layers presented in [1], on which we fine-tuned the two last sibling layers (softmax and bounding box regressor) in order to detect the categories corresponding to visual classes (i.e., *Fruit*, *Leaf*, *Peduncle* and *Petiole*) plus background. We trained the Fast R-CNN with mini-batches of 2 images for 30,000 iterations with a learning rate initialized at 0.001 and reduced by a factor of 10 for the last 10,000 iterations. Training patches were extracted through selective search: positive samples were chosen as those having an intersection over union score with ground truth bounding boxes over 0.5, while the remaining bounding boxes were used as negative samples. Standard random-flipping data augmentation and mini-batch (batch size: 128 ROIs, 64 ROIs sampled for each image) SGD training were employed to train the model. In Table. 5.1 Selective Search and Fast R-CNN performance on the *VegImage* dataset are reported; in particular, for Selective Search object proposals we computed the *ABO* metric (Average Best Overlap per class according to the Pascal Overlap criterion) while

Classes	**Selective Search**	**Fast R-CNN**
Fruit	0.43	35.4
Leaf	0.40	27.1
Peduncle	0.35	26.3
Petiole	0.33	9.7
Mean	0.38	24.5

Table 5.1: *Performance of object proposal and object detection algorithms on the* VegImage *dataset. Selective Search results are reported in terms of* ABO *(Average Best Overlap), whilst the* mAP *(mean Average Precision) is used to compute Fast R-CNN performance.*

Fast R-CNN object detections are evaluated in terms of mAP (mean Average Precision). Even tough both Selective Search and Fast R-CNN algorithms yield low performance on the *VegImage* dataset, the integration in our algorithm of the non-maximum suppression, which leaves only one object for each class, corresponding to the one with the higher likelihood, allow our classifier to work with detections characterized by a very high precision.

As classifier to identify visual properties (colors, shapes, etc.), we employed a pretrained CNN-based classifier (i.e., OverFeat [2]); in particular we reused the first two convolutional layers from the pretrained model to extract a 96-dimensional feature vector provided as input to a neural network, consisting of a fully connected layer followed by a softmax classifier, fine-tuned on a case-by-case basis to detect classes' attributes. In order to train and test these attributes'

Property	*#classes*	**Accuracy**
FruitColor	13	50.9%
FruitOvercolor	17	55.4%
FruitShape	14	35.6%
FruitLenticels	18	49.4%
FruitStripes	4	83.2%
FruitRusset	10	82%
LeafEdge	6	39.8%
LeafShape	4	44.6%

Table 5.2: *Performance of properties' detectors: for each visual attribute we report the number of properties values (i.e., the dimension of classifiers' output) and the detection accuracy.*

detectors we relied on the ground-truth bounding boxes and the properties' annotations available in the *VegImage* dataset. Classification accuracy of each property detector is presented in Table. 5.2, where *#classes* represents the number of values a property can assume; in particular, the recognition of some properties, such as *FruitShape*, *FruitLenticels*, *Leaf Edge* and *LeafShape*, represents a task which can hardly be solved even using deep features.

Semantic-based classifier is combined to visual-based classifier, which in our case, consisted of DenseNet121 model pre-trained on ImageNet [4] and fine-tuned on the *VegImage* dataset. DenseNet fine-tuning was performed by replacing the softmax layer and training all but the first two network layers. Random cropping and horizontal flipping were employed for data augmentation, and mini-batch (batch size: 16) SGD with momentum for training. Learning rate was set to 0.005 with a decay of 0.5. Weight decay was set to 0.00005. We empirically set the parameter α to 0.015.

Given the flexibility of the Bayesian Networks, we evaluated our classification approach with different configurations; Table. 5.3 shows the performance achieved removing one node (corresponding to a visual property or a visual class) at a time in order to identify which ones contribute the most to the classification accuracy; performance was compared to the baseline represented by the DenseNet model fine-tuned on the *VegImage* dataset. Removing the node related to the visual class *Leaf* led to the best performance; this is due to the low average precision achieved by the Fast R-CNN object detector on this visual class together with the low accuracy of the detectors trained to identify the two properties *LeafEdge* and *LeafShape*; higher performance of *Leaf* properties' detectors would likely allow to better exploit this context class in the classification phase.

Graph Structure	**Belief Propagation**	**DenseNet + Belief Propagation**
All nodes	23.8%	47.9%
All nodes \ FruitColor	20.1%	47.3%
All nodes \ FruitOvercolor	23.5%	47.6%
All nodes \ FruitShape	19.5%	46.7%
All nodes \ FruitLenticels	23.6%	47.1%
All nodes \ FruitStripes	21.4%	47.7%
All nodes \ FruitRusset	23.8%	47.9%
All nodes \ LeafEdge	23.6%	47.8%
All nodes \ LeafShape	23.6%	48%
All nodes \ Leaf	24.6%	**48.6%**
DenseNet	44.2%	

Table 5.3: *Performance of the proposed classification approach: each row corresponds to a different configuration of the Bayesian Network graph structure. We report the accuracy obtained when using the Belief Propagation inference only and when beliefs are combined with the fine-tuned DenseNet model output. Performance reached using only visual features are represented by the DenseNet accuracy. Best performance achieved removing the* Leaf *node from the graph.*

Furthermore, we evaluated how the overall classification performance changed w.r.t the distribution of properties detectors' likelihoods. In particular we assessed the classification performance by imposing a uniform misclassification rate on the detectors' outputs. Figure 5.3 shows the performance obtained by Belief Propagation inference over the data simulated at different error values; it can be noted that a 10% uniform error on visual properties does not cause a 10% drop of the overall classification accuracy, likewise when the uniform error is set to 20% or 30% Belief Propagation achieved a classification accuracy higher than 80% or 70%. Moreover, when we forced a misclassification of 45 visual properties over 100 (which corresponds more or less to the average error obtained by the properties' detectors as reported in Table 5.2), the Belief Propagation accuracy obtained was around 67%, about 20% higher than the best result presented in Table 5.3; this gap is due to the error distribution in the *VegImage* dataset, which, of course, is not uniformly distributed as supposed in the simulation. Indeed, the classification of some properties, such as *Yellow* color or *Absent* stripes, is much easier if compared to the classification of property values whose recognition represents a very complex task even for domain experts (e.g., *GoldenYellow* color or *SlightlyVisible* stripes). Nevertheless, the performed simulation provides a basis to assess the validity of our semantic classifier, whose performance are substantially affected by the visual properties' detectors and by the bias existing in the dataset.

We finally compared our semantic-based classification approach – described in Sect.5.1 – to common state-of-the-art fine-grained classification methods, namely OverFeat [97] GoogLeNet [4], ResNet [127]

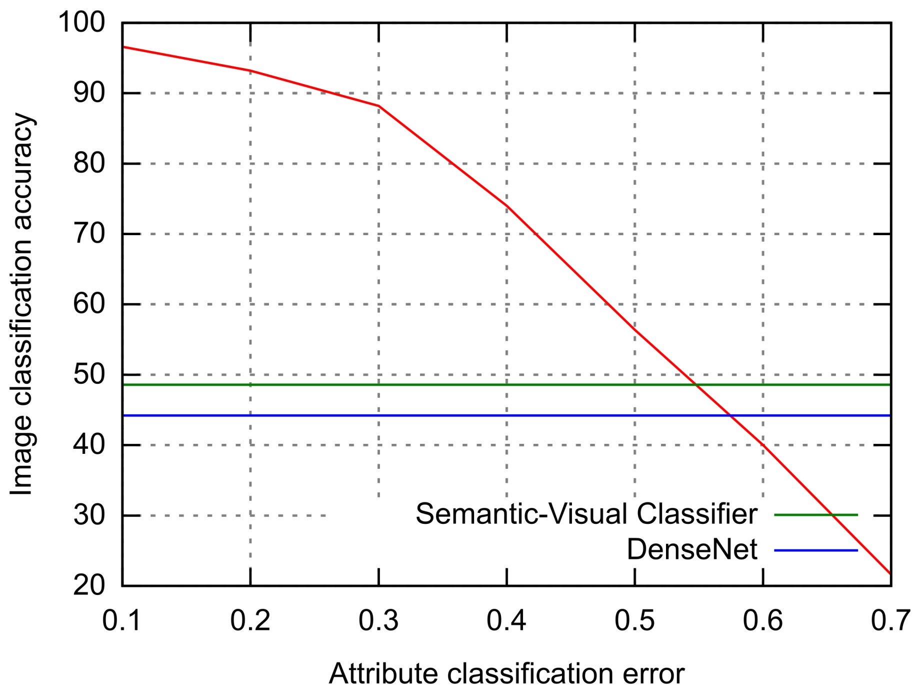


Figure 5.3: *In red the classification accuracy obtained by Belief Propagation inference over the simulated data obtained varying the rate of the misclassification error imposed to the visual properties' detectors. For error percentage lower than 50%, Belief Propagation performance on simulated data are higher than the ones achieved by the combination of visual classifiers with the marginals computed applying the properties' detectors to the real data (green line) or the ones obtained using visual descriptors only (blue line), as reported in Table 5.3.*

and DenseNet [128], pre-trained on ImageNet and fine-tuned on the *VegImage* dataset. Training settings for Overfeat, ReseNet and

Learning Visual Descriptors			
OverFeat	**GoogLeNet**	**ResNet**	**DenseNet**
26.6%	41.5%	41.1%	44.2%
Exploiting High-Level Knowledge			
	GoogleNet + BP	**ResNet + BP**	**DenseNet + BP**
—	44.8%	44.5%	48.6%

Table 5.4: *Mean Classification Accuracy achieved a) when using only low and middle-level visual descriptors (top row) and b) when integrating high-level knowledge (bottom row).*

GoogleNet were set the same as DenseNet fine-tuning for integration with knowledge-based classifier (see above). The performance in terms of mean classification accuracy is presented in Table 5.4, which reports results obtained learning visual descriptors only along with the improved accuracy achieved when integrating high level knowledge in the classification process; our semantic-based method yields an average performance increase (with respect to the DenseNet baseline) of about 4.4%. This provides evidence of our original claim, i.e., that by adding semantic knowledge it is possible to improve performance w.r.t. learning visual features only . Indeed, the fruit dataset describes an application domain where class discrimination is strongly based on a context dependency between objects, which needs to be encoded and integrated into the classification methods as *a priori* information.

5.3 Discussion

In this Chapter we demonstrated that the integration of visual world semantics into computational models supports greatly computer vision methods. In particular, we proposed a new classifier able to exploit jointly semantic high-level knowledge and visual cues. The classifier was tested on the *VegImage* dataset, described in Chapter 4 and the experimental results achieved on the fruit classification task demonstrates that integrating knowledge-enriched visual annotations outperforms state-of-the-art fine-grained and deep learning approaches. Results also suggest that deep-learning show difficulties in tackling highly-specialized tasks, for which human knowledge in particularly needed and for which it is not easy to collect large - annotated - datasets.

5.4 Publications

- Francesca Murabito, Simone Palazzo, Concetto Spampinato, and Daniela Giordano. Exploiting structured high-level knowledge for domain-specific visual classification. In *PAMI, under (second) review*, 2019..

CHAPTER

SIX

CONCLUSIONS

In this thesis we explored several ways to incorporate human capabilities into automatic models, from the computation of saliency maps guided by a classification task to the creation of semantic-enriched datasets to the development of semantic-driven deep neural networks trained to perform fine-grained visual recognition.

Initially, we introduced the *SalClassNet* model, an end-to-end convolutional neural network which computes saliency maps, by emulating the way humans shift their attention accordingly to the task to be performed, and exploits those maps to guide a classification network. To test this approach, we collected a dataset of eye-gaze maps by asking several subjects to look at images from the Stanford Dogs dataset, with the objective of distinguishing dog breeds. Performance analysis on our dataset and other saliency benchmarking datasets showed that *SalClassNet* outperforms state-of-the-art saliency detectors, such as SalNet and SALICON. Finally, we also analyzed the performance of our model in a fine-grained recognition task and found out that it yields enhanced classification accuracy compared to GoogleNet and VGG-19 classifiers. The achieved results, thus, demonstrate that 1) conditioning saliency detectors with object classes reaches state-of-the-art performance, and 2) explicitly providing top-down saliency maps to visual classifiers enhances accuracy.

To further investigate how integrating human skills to computational models enhance performance we attempted to exploit high-level visual knowledge for fine-grained visual categorization.

Nevertheless, incorporating high-level knowledge into classic machine learning methods poses several unexplored challenges from its extraction and modeling to its effective inclusion into classification approaches. To address these challenges, we employed computational ontologies to a) model visual knowledge and b) built a hybrid visual-semantic classification framework. Our classification method is based on building a Bayesian Network grap,h whose structure depends on the knowledge encoded in the ontology, and performing an inference over the graph, which backs up a standard CNN classifier.

We tested our approach on the fruit variety classification task. To this end, we built *VegImage*, a dataset with 3,872 images from 24 different fruit varieties (with only subtle visual appearance differences), over 60,000 bounding boxes of fruits, leaves, etc. and a large knowledge base (1 million OWL triples) representing a-priori knowledge about object appearance. Performance analysis showed that our method improve state-of-the-art purely-visual classifiers, substantiating our claims.

All the approaches presented demonstrated how the development of algorithms that emulate the whole complex of perceptive and cognitive human processes and which are not restricted to the trivial reproduction of the results through a "black-box" models, such as convolutional neural networks, can effectively help in achieving higher performance in several computer vision tasks.

BIBLIOGRAPHY

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS*, 2012.

[2] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015.

[5] L. Itti and C. Koch, "A saliency-based search mechanism for overt and covert shifts of visual attention," *Vision Research 2000*, pp. 1489–1506, 2000.

[6] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *ICVGIP*, 2008.

[7] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and Dogs," in *CVPR*, 2012.

[8] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," no. CNS-TR-2011-001, 2011.

[9] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov, "DeViSE: A Deep Visual-Semantic Embedding Model," in *NIPS*, 2013.

[10] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks," in *CVPR*, 2014.

[11] C. Lampert, H. Nickisch, and S. Harmeling, "Attribute-Based Classification for Zero-Shot Visual Object Categorization," *PAMI*, 2014.

[12] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-Scale Object Classification Using Label Relation Graphs," in *ECCV*, 2014.

[13] V. Ramanathan, C. Li, J. Deng, W. Han, Z. Li, K. Gu, Y. Song, S. Bengio, C. Rossenberg, and L. Fei-Fei, "Learning semantic relationships for better action retrieval in images," in *CVPR*, 2015.

[14] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie, "Visual recognition with humans in the loop," in *ECCV*, 2010.

[15] J. Deng, J. Krause, and L. Fei-Fei, "Fine-grained crowdsourcing for fine-grained recognition," in *CVPR*, 2013.

[16] G. Li and Y. Yu, "Visual saliency detection based on multiscale deep CNN features," *Transactions on Image Processing 2016*, pp. 5012–5024, 2016.

[17] M. Kümmerer, L. Theis, and M. Bethge, "Deep Gaze I: Boosting saliency prediction with feature maps trained on imagenet," in *ICLRW*, 2015.

[18] X. Huang, C. Shen, X. Boix, and Q. Zhao, "Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks," in *ICCV*, 2015.

[19] J. Pan, E. Sayrol, X. Giro-i Nieto, K. McGuinness, and N. E. O'Connor, "Shallow and deep convolutional networks for saliency prediction," in *CVPR*, 2016.

[20] S. He, R. W. H. Lau, W. Liu, Z. Huang, and Q. Yang, "SuperCNN: A superpixelwise convolutional neural network for salient object detection," *IJCV*, 2015.

[21] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu, "Predicting eye fixations using convolutional neural networks," in *CVPR*, 2015.

[22] G. Li and Y. Yu, "Visual saliency based on multiscale deep features," in *CVPR*, 2015.

[23] R. Zhao, W. Ouyang, H. Li, and X. Wang, "Saliency detection by multi-context deep learning," in *CVPR*, 2015.

[24] J. Han, D. Zhang, S. Wen, L. Guo, T. Liu, and X. Li, "Two-stage learning to predict human eye fixations via sdaes," *Transaction on Cybernetics*, 2016.

[25] R. J. Peters and L. Itti, "Beyond bottom-up: Incorporating task-dependent influences into a computational model of spatial attention," in *CVPR*, 2007.

[26] T. Judd, K. Ehinger, F. Durand, and A. Torralba, "Learning to predict where humans look," in *ICCV*, 2009.

[27] L. Itti, "Probabilistic learning of task-specific visual attention," in *CVPR*, 2012.

[28] G. Zhu, Q. Wang, and Y. Yuan, "Tag-saliency: Combining bottom-up and top-down information for saliency detection," *CVIU*, 2014.

[29] Y. Lin, S. Kong, D. Wang, and Y. Zhuang, "Saliency detection within a deep convolutional architecture," in *AAAIW*, 2014.

[30] C. Shen and Q. Zhao, "Learning to predict eye fixations for semantic contents using multi-layer sparse network," *Neurocomputing*, 2014.

[31] L. Wang, H. Lu, X. Ruan, and M. H. Yang, "Deep networks for saliency detection via local estimation and global search," in *CVPR*, 2015.

[32] Y. Tang and X. Wu, "Saliency detection via combining region-level and pixel-level predictions with cnns," in *ECCV*, 2016.

[33] T. Chen, L. Lin, L. Liu, X. Luo, and X. Li, "Disc: Deep image saliency computing via progressive representation learning," *Transaction on NNLS*, 2016.

[34] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *CVPR*, 2016.

[35] C. Cao, X. Liu, Y. Yang, Y. Yu, J. Wang, Z. Wang, Y. Huang, L. Wang, C. Huang, W. Xu, *et al.*, "Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks," in *CVPR*, 2015.

[36] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," in *ECCV*, 2016.

[37] M. Jiang, S. Huang, J. Duan, and Q. Zhao, "Salicon: Saliency in context," in *CVPR*, 2015.

[38] P. Xu, K. A. Ehinger, Y. Zhang, A. Finkelstein, S. R. Kulkarni, and J. Xiao, "Turkergaze: crowdsourcing saliency with webcam based eye tracking," *arXiv preprint arXiv:1504.06755*, 2015.

[39] Z. Bylinskii, T. Judd, A. Borji, L. Itti, F. Durand, A. Oliva, and A. Torralba, "Mit saliency benchmark."

[40] M. Kümmerer, T. S. Wallis, and M. Bethge, "DeepGaze II: Reading fixations from deep features trained on object recognition," *arXiv preprint arXiv:1610.01563*, 2016.

[41] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, “Multi-level net: A visual saliency prediction model,” in *ECCVW*, 2016.

[42] S. S. Kruthiventi, K. Ayush, and R. V. Babu, “Deepfix: A fully convolutional neural network for predicting human eye fixations,” *IP*, 2017.

[43] E. Vig, M. Dorr, and D. Cox, “Large-scale optimization of hierarchical features for saliency prediction in natural images,” in *CVPR*, 2014.

[44] S. Jetley, N. Murray, and E. Vig, “End-to-end saliency mapping via probability distribution prediction,” in *CVPR*, 2016.

[45] S. Ebert, D. Larlus, and B. Schiele, “Extracting structures in image collections for object recognition.,” in *ECCV*, 2010.

[46] T. Malisiewicz and A. A. Efros, “Beyond Categories: The Visual Memex Model for Reasoning About Object Relationships,” in *NIPS*, 2009.

[47] G. Patterson and J. Hays, “SUN Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes,” in *CVPR*, 2012.

[48] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, “Objects in Context,” in *ICCV*, 2007.

[49] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, “Learning hierarchical models of scenes, objects, and parts,” in *ICCV*, 2005.

[50] C. Galleguillos and S. Belongie, "Context based object categorization: A critical survey," *CVIU*, 2010.

[51] C. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, 2009.

[52] D. Parikh and K. Grauman, "Relative attributes," in *ICCV*, 2011.

[53] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *CVPR*, 2009.

[54] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *PAMI*, 2004.

[55] D. Kuettel, M. Guillaumin, and V. Ferrari, "Segmentation Propagation in ImageNet," in *ECCV*, 2012.

[56] A. Torralba, "Contextual priming for object detection," in *IJCV*, 2003.

[57] Y. Liu, R. Wang, S. Shan, and X. Chen, "Structure inference net: Object detection using scene-level context and instance-level relationships," in *CVPR*, 2018.

[58] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for attribute-based classification," in *CVPR*, 2013.

[59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *CVPR*, 2009.

[60] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.

[61] M. Fisher, M. Savva, and P. Hanrahan, "Characterizing structural relationships in scenes using graph kernels," *ACMTG*, 2011.

[62] J. Yang, B. Price, S. Cohen, and M.-H. Yang, "Context driven scene parsing with attention to rare classes," in *CVPR*, 2014.

[63] D. Lin, S. Fidler, C. Kong, and R. Urtasun, "Visual semantic search: Retrieving videos via complex textual queries," in *CVPR*, 2014.

[64] J. Johnson, R. Krishna, M. Stark, L.-j. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-fei, "Image Retrieval using Scene Graphs," in *CVPR*, 2015.

[65] N. Maillot, M. Thonnat, and A. Boucher, "Towards ontology-based cognitive vision," *MVA*, 2004.

[66] C. Wah, S. Branson, P. Perona, and S. Belongie, "Multiclass recognition and part localization with humans in the loop," in *ICCV*, 2011.

[67] D. S. Cheng, F. Setti, N. Zeni, R. Ferrario, and M. Cristani, "Semantically-driven automatic creation of training sets for object recognition," *CVIU*, 2015.

[68] S. Dasiopoulou, E. Giannakidou, and G. Litos, "A Survey of Semantic Image and Video Annotation Tools," *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*, 2011.

[69] C. Halaschek-Wiener, J. Golbeck, A. Schain, M. Grove, B. Parsia, and J. A. Hendler, "PhotoStuff-An Image Annotation Tool for the Semantic Web," in *ISWC*, 2005.

[70] K. Petridis, D. Anastasopoulos, C. Saathoff, N. Timmermann, Y. Kompatsiaris, and S. Staab, "M-OntoMat-Annotizer: Image annotation linking ontologies and multimedia low-level features," *KES*, 2006.

[71] F. Murabito, S. Palazzo, C. Spampinato, and D. Giordano, "Generating knowledge-enriched image annotations for fine-grained visual classification," in *ICIAP*, 2017.

[72] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *PRL*, 2009.

[73] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "TextonBoost for Image Understanding: Multi-Class Object Recognition and Segmentation by Jointly Modeling Texture, Layout, and Context," in *IJCV*, 2009.

[74] N. Crofts, M. Doerr, and T. Gill, *The CIDOC Conceptual Reference Model: a standard for communication cultural contents.* 2003.

[75] T. Stasinopoulou, L. Bountouri, C. Kakali, I. Lourdi, C. Papatheodorou, M. Doerr, and M. Gergatsoulis, "Ontology-based metadata integration in the cultural heritage domain," in *International Conference on Asian Digital Libraries*, 2007.

[76] D. A. Koutsomitropoulos and T. S. Papatheodorou, "Expressive reasoning about cultural heritage knowledge using web ontologies.," in *WEBIST (2)*, 2007.

[77] V. Alexiev, D. Manov, J. Parvanova, S. Petrov, *et al.*, "Large-scale reasoning with a complex cultural heritage ontology (cidoc crm)," *A Mapping of CIDOC CRM Events to German Wordnet for Event Detection in Texts*, 2013.

[78] C. Ghiselli, L. Bozzato, and A. Trombetta, "Representation and management of ontologies in cultural heritage domain.," in *SWAP*, 2005.

[79] L. Bing, K. C. Chan, and L. Carr, "Using aligned ontology model to convert cultural heritage resources into semantic web," in *ICSC*, 2014.

[80] G. Schreiber, A. Amin, L. Aroyo, M. van Assem, V. de Boer, L. Hardman, M. Hildebrand, B. Omelayenko, J. van Osenbruggen, A. Tordai, *et al.*, "Semantic annotation and search of

cultural-heritage collections: The multimedian e-culture demonstrator," *Web Semantics: Science, Services and Agents on the World Wide Web*, 2008.

[81] M. Koolen, J. Kamps, and V. de Keijzer, "Information retrieval in cultural heritage," *Interdisciplinary Science Reviews*, 2009.

[82] J. S. Hare, P. A. Sinclair, P. H. Lewis, K. Martinez, P. G. Enser, and C. J. Sandom, "Bridging the semantic gap in multimedia information retrieval: Top-down and bottom-up approaches," 2006.

[83] J. Llamas, P. M. Lerones, E. Zalama, and J. Gómez-García-Bermejo, "Applying deep learning techniques to cultural heritage images within the inception project," in *Euro-Mediterranean Conference*, 2016.

[84] D. Walther, U. Rutishauser, C. Koch, and P. Perona, "Selective visual attention enables learning and recognition of multiple objects in cluttered scenes," *CVIU*, 2005.

[85] Z. Ren, S. Gao, L.-T. Chia, and I. W.-H. Tsang, "Region-based saliency detection and its application in object recognition," in *TCSVT*, 2015.

[86] X. Zhang, H. Xiong, W. Zhou, W. Lin, and Q. Tian, "Picking deep filter responses for fine-grained image recognition," in *CVPR*, 2016.

[87] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," in *ICLR*, 2015.

[88] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu, "Recurrent models of visual attention," in *NIPS*, 2014.

[89] A. Almahairi, N. Ballas, T. Cooijmans, Y. Zheng, H. Larochelle, and A. Courville, "Dynamic capacity networks," in *ICML*, 2016.

[90] A. Oliva and A. Torralba, "The role of context in object recognition," *TICS*, 2007.

[91] L. Xie, Q. Tian, R. Hong, S. Yan, and B. Zhang, "Hierarchical part matching for fine-grained visual categorization," in *ICCV*, 2013.

[92] L. Bo, X. Ren, and D. Fox, "Kernel descriptors for visual recognition," in *NIPS*, 2010.

[93] S. Yang, L. Bo, J. Wang, and L. G. Shapiro, "Unsupervised template learning for fine-grained object recognition," in *NIPS*, 2012.

[94] T. Berg and P. N. Belhumeur, "Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in *CVPR*, 2013.

[95] N. Zhang, R. Farrell, F. Iandola, and T. Darrell, "Deformable part descriptors for fine-grained recognition and attribute prediction," in *ICCV*, 2013.

[96] E. Gavves, B. Fernando, C. Snoek, A. Smeulders, and T. Tuytelaars, "Fine-grained categorization by alignments," in *ICCV*, 2013.

[97] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *CVPRW*, 2014.

[98] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," in *ICML*, 2014.

[99] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnns for fine-grained visual recognition," in *PAMI*, 2017.

[100] S. Branson, G. Van Horn, S. Belongie, and P. Perona, "Bird species categorization using pose normalized deep convolutional nets," *arXiv preprint arXiv:1406.2952*, 2014.

[101] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based R-CNNs for fine-grained category detection," in *ECCV*, 2014.

[102] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *NIPS*, 2015.

[103] H. Zhang, T. Xu, M. Elhoseiny, X. Huang, S. Zhang, A. Elgammal, and D. Metaxas, "Spda-cnn: Unifying semantic part detection and abstraction for fine-grained recognition," in *CVPR*, 2016.

[104] M. Simon and E. Rodner, "Neural activation constellations: Unsupervised part model discovery with convolutional networks," in *ICCV*, 2015.

[105] T. Chen, L. Lin, R. Chen, Y. Wu, and X. Luo, "Knowledge-embedded representation learning for fine-grained image recognition," *IJCAI*, 2018.

[106] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *ICLR*, 2016.

[107] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei, "Novel dataset for fine-grained image categorization," in *CVPRW*, 2011.

[108] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2014.

[109] J. T. Enns and S. C. MacDonald, "The role of clarity and blur in guiding visual attention in photographs," *J Exp Psychol Hum Percept Perform*, 2013.

[110] D. P. Papadopoulos, A. D. Clarke, F. Keller, and V. Ferrari, "Training object class detectors from eye tracking data," in *ECCV*, 2014.

[111] A. Borji and L. Itti, "Cat2000: A large scale fixation dataset for boosting saliency research," *CVPRW*, 2015.

[112] Z. Bylinskii, P. Isola, C. Bainbridge, A. Torralba, and A. Oliva, "Intrinsic and extrinsic effects on image memorability," *Vision research*, 2015.

[113] M. Jiang, J. Xu, and Q. Zhao, "Saliency in crowd," in *ECCV*, 2014.

[114] J. Xu, M. Jiang, S. Wang, M. S. Kankanhalli, and Q. Zhao, "Predicting human gaze beyond pixels," *JoV*, 2014.

[115] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, "The secrets of salient object segmentation," in *CVPR*, 2014.

[116] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *PAMI*, 2013.

[117] A. Borji, D. N. Sihite, and L. Itti, "Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study," *TIP*, 2013.

[118] Y.-S. Wong, H.-K. Chu, and N. J. Mitra, "SmartAnnotator An Interactive Tool for Annotating Indoor RGBD Images," *CGF*, 2015.

[119] I. Kavasidis, S. Palazzo, R. Salvo, D. Giordano, and C. Spampinato, "An innovative web-based collaborative platform for video annotation," *MTAP*, 2014.

[120] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and web-based tool for image annotation," in *IJCV*, 2008.

[121] M. Horridge and S. Bechhofer, "The OWL API: A Java API for OWL Ontologies," *SWJ*, 2011.

[122] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," *arXiv preprint arXiv*, 2014.

[123] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie, "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in *CVPR*, 2015.

[124] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding Belief Propagation and Its Generalizations. Exploring Artificial Intelligence in the New Millennium.," 2003.

[125] J. Chua and P. F. Felzenszwalb, "Scene grammars, factor graphs, and belief propagation," *CoRR*, 2016.

[126] R. Girshick, "Fast R-CNN," in *ICCV*, 2015.

[127] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[128] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks.," in *CVPR*, 2017.